



José Manuel
Clemente dos Santos

**SISTEMA DE CONTROLO DE IRRIGAÇÃO BASEADO
EM LINGUAGEM JAVA**



**José Manuel
Clemente dos Santos** **SISTEMA DE CONTROLO DE IRRIGAÇÃO BASEADO
EM LINGUAGEM JAVA**

Dissertação apresentada à Universidade de Aveiro para cumprimento dos requisitos necessários à obtenção do grau de Mestre em Engenharia Electrónica e Telecomunicações, realizada sob a orientação científica do Dr. Alexandre Mota, Professor Associado do Departamento de Electrónica, Telecomunicações e Informática da Universidade de Aveiro

O júri

Presidente

Prof. Dr. José Alberto Gouveia Fonseca

Professor Associado da Universidade de Aveiro

Vogais

Prof. Dr. José António Barros Vieira

Professor Adjunto do Instituto Politécnico de Castelo Branco

Prof. Dr. Alexandre Manuel Moutela Nunes da Mota (Orientador)

Professor Associado da Universidade de Aveiro

Agradecimentos

É a primeira atitude a ter com todos os que me incentivaram, acompanharam e ajudaram na realização deste trabalho:

A Deus, pelas faculdades físicas e talento que me concedeste no decorrer de todo o curso;

À minha família, pelo apoio incondicional e compreensão que sempre tiveram para comigo;

Ao Professor Doutor Alexandre Manuel Mota, pela orientação científica e disponibilidade sempre demonstradas;

A todos os docentes, que ao longo desta caminhada me deram o melhor de si;

Ao DETI - Universidade de Aveiro, que forneceu condições para a realização deste projecto;

Aos colegas e amigos, que contribuíram para que todo o tempo do curso fosse mais agradável e me apoiaram nos bons e maus momentos.

Palavras-chave

Sistemas de irrigação, Java, Barramento *I-Wire*, controlo difuso

Resumo

A gestão da irrigação consiste em determinar o tempo e o momento certo de funcionamento de um equipamento de irrigação, por forma a aplicar a quantidade de água necessária para o pleno desenvolvimento da cultura. A automação dos sistemas de irrigação permite controlar estes sistemas de forma mais simples e precisa. No presente trabalho propõe-se divulgar um sistema de controlo de irrigação baseado em linguagem Java. Este sistema pretende ser mais económico fazendo uma melhor gestão dos recursos e proporciona uma maior mobilidade dos seus utilizadores, uma vez que pode ser acedido remotamente.

Keywords

Irrigation systems, Java, 1-Wire bus, Fuzzy control

Abstract

The irrigation management determines the right time and moment to operate an irrigation equipment in order to apply the amount of water needed for the full development of the culture. The irrigation systems automation, allows to control the systems in a more simply and more accurately way. In this work it is proposed, to divulge a system of irrigation based on Java. This system aims to be cheaper with a better management of resources and provides greater mobility to their users, since it can be accessed remotely.

Conteúdo

1	Introdução	1
1.1	Perspectiva Histórica	1
1.2	Motivação	3
1.3	Objectivos	3
1.4	Pressupostos	3
1.5	Estrutura da Dissertação	3
2	Irrigação	5
2.1	Interacções da irrigação com o solo, plantas, clima e água	5
2.1.1	Solo	5
2.1.2	Plantas	6
2.1.3	Clima	6
2.1.4	Água	6
2.2	Métodos de Irrigação	6
2.2.1	Superfície ou Gravidade	7
2.2.1.1	Rega por sulco	7
2.2.1.2	Rega por faixa	7
2.2.1.3	Rega por inundação	8
2.2.2	Localizada	8
2.2.2.1	Rega gota-a-gota	8
2.2.2.2	Rega por microaspersão	9
2.2.3	Aspersão	10
2.2.3.1	Canhão hidráulico	11
2.2.3.2	Pivô	11
2.3	Sistemas de controlo de Irrigação	12
2.3.1	Controladores de irrigação por horas	12
2.3.2	Controladores de irrigação baseados em factores meteorológicos	13
2.4	Soluções Comerciais	14

3	Sensores e Actuadores	17
3.1	Sensores de Humidade do Solo	17
3.1.1	Resistivos	18
3.1.2	Capacitivos	19
3.1.3	Tensiómetros	20
3.1.4	Sonda de Neutrões	21
3.1.5	TDR (Reflectometria no Domínio do Tempo)	22
3.1.6	DPHP (<i>Dual-Probe Heat-Pulse</i>)	22
3.2	Sensores de Pluviosidade	23
3.2.1	Sensor de chuva	23
3.2.2	Medidor de chuva	24
3.3	Outros Sensores	24
3.4	Electroválvulas	26
4	Sistemas Embutidos Baseados em Linguagem Java	29
4.1	Linguagem Java	29
4.1.1	Máquina Virtual Java	30
4.1.2	Vantagens e desvantagens	31
4.1.3	Alguns Sistemas embutidos baseados em Java	32
4.2	Plataforma TINI	33
4.2.1	TINI400	33
4.2.1.1	DSTINIm400	33
4.2.1.2	DSTINIs400	35
4.2.2	Sistema Operativo TINI	36
4.2.3	Máquina Virtual Java e API	37
4.3	Barramento <i>1-Wire</i>	38
4.3.1	Transmissão de dados	38
4.3.2	Alimentação dos Dispositivos	40
4.3.2.1	Alimentação Parasita	41
4.3.2.2	Alimentação Externa	41
4.3.3	Topologias das redes <i>1-Wire</i>	41
4.3.3.1	Topologia linear	41
4.3.3.2	Topologia ramificada	42
4.3.3.3	Topologia em estrela	42
4.3.4	Tipos de dispositivos	42
4.3.5	Avaliação de Desempenho de <i>1-Wire</i>	43

5	Controlo Difuso	45
5.1	Fuzificação	46
5.2	Inferência	46
5.2.1	Base de Regras	46
5.2.2	Mecanismo de inferência	47
5.3	Desfuzificação	48
5.4	Controlo Difuso em Sistemas Embutidos	48
6	Implementação	49
6.1	Sensores	49
6.1.1	Sensor de Temperatura	50
6.1.2	Sensor de Humidade do ar	50
6.1.3	Sensor de Humidade do solo	51
6.1.4	Sensor de Radiação solar	53
6.1.5	Medidor de Chuva	54
6.2	Actuador	55
6.3	Interface Local	56
6.4	Interface Remoto	58
7	Resultados e conclusões	61
7.1	Caracterização do sensor de humidade do solo	62
7.1.1	Transição do estado seco para o estado húmido	62
7.1.2	Transição de estado húmido para seco	62
7.1.3	Sensibilidade ao factor temperatura	63
7.2	Sistema de controlo	65
7.2.1	Estrutura do sistema	66
7.2.2	Algoritmo de controlo	67
7.2.3	<i>Software</i> presente na plataforma TINI	69
7.2.4	Resultados	70
8	Conclusões e Trabalho Futuro	73
8.1	Conclusões	73
8.2	Trabalho futuro	73
	Bibliografia	75
A	Lógica Difusa	79
B	Instalação de <i>Software</i> TINI	87

C Esquema Eléctrico do Interface Local**93**

Lista de Figuras

2.1	Rega por sulco	7
2.2	Rega por faixa	8
2.3	Rega por inundação	8
2.4	Gotejamento	9
2.5	Microaspersores	9
2.6	Disposição em quadrado	10
2.7	Disposição em rectângulo	10
2.8	Disposição em triângulo	10
2.9	Canhão Hidráulico	11
2.10	Pivô	12
2.11	Controlo em malha aberta	12
2.12	Controlo de malha fechada	13
3.1	Curva de calibração Watermark (Humidade do solo em KPa e Resistência em ohms)	18
3.2	Circuito de sensor resistivo	18
3.3	Sensores resistivos de humidade do solo	19
3.4	Sensor capacitivo e Circuito oscilador	20
3.5	Sensor Capacitivo	20
3.6	Tensiómetros	21
3.7	Sonda de Neutrões	21
3.8	Sensor TDR	22
3.9	DPHP	23
3.10	Sensor de chuva	23
3.11	Medidor de chuva	24
3.12	Sensor de temperatura RTD	25
3.13	Sensor de humidade do ar	25
3.14	Painel solar	25

3.15	Velocidade do vento	26
3.16	Estação meteorológica	26
3.17	Electroválvulas	27
3.18	Electroválvula por pulso [1]	28
4.1	Compilar em Java[2]	30
4.2	Independência do Sistema Operativo em Java [2]	30
4.3	DSTINIm400	35
4.4	DSTINIs400	36
4.5	TINI OS	37
4.6	Barramento <i>1-Wire</i>	39
4.7	Níveis lógicos CMOS/TTL	39
4.8	Escrever zero ou um em <i>1-wire</i>	40
4.9	Sequência de transmissão <i>1-Wire</i>	40
4.10	Topologia linear	41
4.11	Topologia ramificada	42
4.12	Topologia em estrela	42
5.1	Controlador difuso	45
5.2	Universo de discurso e funções de pertença	46
5.3	Representação gráfica do mecanismo de inferência <i>max-min</i>	47
5.4	Representação gráfica do mecanismo de inferência <i>max-prod</i>	48
6.1	Sistema de controlo de irrigação	49
6.2	Sensor de Temperatura	50
6.3	Sensor de humidade do ar implementado	50
6.4	Sensor de humidade do solo implementado	51
6.5	Esquema eléctrico do sensor de humidade do solo	52
6.6	Fotografia do circuito e sensor de humidade	52
6.7	Sensor de radiação solar (Diagrama de blocos)	53
6.8	Sensor de radiação solar (Esquema eléctrico)	53
6.9	Sensor de radiação solar (Fotografia)	54
6.10	Medidor de chuva	54
6.11	Balde afunilado e o balancé com um <i>reed switch</i>	55
6.12	Circuito implementado para o actuador	55
6.13	Módulo de potência do actuador	56
6.14	Interface local	56
6.15	Diagrama do código do interface local	57

6.16	Fotografias do interface local e circuito eléctrico	58
6.17	Interface remoto	58
6.18	Plataforma gráfica no MATLAB	59
7.1	Teste Watermark sensor seco para húmido	62
7.2	Teste Watermark sensor húmido para seco	63
7.3	Comparação da resistência medida sem compensação e com compensação	64
7.4	Temperatura medida ao longo do teste	64
7.5	Comparação da humidade não compensada com as duas compensações	65
7.6	Plataforma de Teste	66
7.7	Fotografia da Plataforma de teste	66
7.8	Conjuntos difusos do solo	67
7.9	Conjuntos difusos de período de rega	68
7.10	Regras difusas	68
7.11	Relação entrada saída algoritmo difuso	68
7.12	<i>Software</i> implementado na TINI	70
7.13	Humidade do solo - teste	70
7.14	Humidade do solo - teste (<i>zoom</i>)	71
A.1	Qualificação de “pessoas altas” segundo conjunto clássico (também conhecido como crisp) e conjunto difuso [3].	81
A.2	Exemplo de funções de permanência μ_A (a)trapezoidal, (b)triangular, (c)trapezoidal suave, (b)triangular suave [3]	82
A.3	União, intersecção e complemento. (a-c) conjuntos clássicos, (d-f) conjuntos difusos	84
B.1	Propriedades do sistema	88
B.2	Variáveis de ambiente	88
B.3	JavaKit	89
B.4	Carregar firmware pelo JavaKit	90
B.5	Firmware 1.17	91
C.1	Esquema eléctrico do interface local	93

Lista de Tabelas

4.1	Comparação de sistemas embutidos baseados em Java [4]	34
7.1	Tabela difusa	69

Capítulo 1

Introdução

1.1 Perspectiva Histórica

No início da civilização a luta pela sobrevivência dos nossos antepassados foi muito complicada, pois não tinham instrumentos nem técnicas para produzir a sua própria alimentação. Para assegurarem a subsistência os homens limitaram-se a utilizar os recursos naturais. Eram, por isso, nómadas e dominava uma economia recolectora.

Com o passar do tempo, descobriram que se recolhessem algumas sementes e as espalhassem pela terra elas voltavam a nascer e, portanto, já não haveria necessidade de andar, constantemente, a mudar de lugar e de recolector o Homem tornou-se produtor e, de nómada, passou a sedentário. Tendo-se apercebido da fertilidade quase contínua começou a praticar a agricultura e passou, assim, a viver nas mesmas regiões com os recursos naturais por ele produzidos numa economia de subsistência.

Depressa descobriram a necessidade que havia de regar as plantas. Assim surgiu a irrigação que, desde logo, veio associada à agricultura e ao desenvolvimento económico dos povos. Nalgumas zonas mais áridas, a agricultura só era possível com a utilização da irrigação. As regiões áridas e semi-áridas foram as primeiras a serem povoadas em toda a história humana e em muitas delas ergueram-se ricos impérios e majestosas civilizações que forjaram a moderna cultura ocidental.

O Egipto é um bom exemplo dessas civilizações pois o rio Nilo exerceu uma importância fundamental no desenvolvimento desse povo oferecendo água e terra cultivável a uma região situada em pleno deserto. Era no entanto, preciso utilizar a inundação, distribuir a água equitativamente, aumentar a superfície irrigada e drenar pântanos. Assim surgiu, no antigo Egipto, a primeira obra de engenharia ligada à irrigação com a construção de diques, represas e canais que melhoravam o aproveitamento das águas do rio. Desde muito cedo, vários foram os povos que surgiram à beira dos rios e, aí, se

fixaram dando origem aos primeiros aglomerados populacionais, que se desenvolveram graças a uma eficaz utilização dos recursos hídricos.

Até ao século XIX, a irrigação foi feita sem equipamentos específicos e, praticamente, só se utilizavam os métodos de inundação e de sulcos. Com a evolução dos tempos, tornou-se evidente que o desenvolvimento económico e a modernização das actividades produtivas eram indispensáveis para o progresso da humanidade. Para tornar as actividades produtivas rentáveis e garantir a competitividade era pertinente revolucionar os sistemas de irrigação.

A grande evolução da irrigação surgiu com a produção de tubulações específicas e estudos de aplicação da água. Na década de 60, surgiu em Israel a irrigação localizada. A invenção do primeiro aspersor de impacto, feita por Orton Englehart, em 1993, foi um facto de grande importância para o mundo da irrigação, pois veio dar início a uma nova era na irrigação mundial [5].

Pela necessidade iminente de economia de água, recurso cada vez mais escasso, esta evolução não parou por aqui. Em busca de um aproveitamento cada vez mais eficaz dos recursos hídricos, o homem tem vindo a desenvolver diversos sistemas de irrigação.

Actualmente existem já sistemas de irrigação bastante desenvolvidos. O surgimento dos sensores foi muito importante, pois permitiu fazer o controlo da irrigação. Existem sistemas de irrigação bastante desenvolvidos, que funcionam por horas. Existem, também, outros ainda mais eficientes com programadores baseados em factores meteorológicos.

Para um melhor aproveitamento de água e minimizar os impactos ambientais provocados pelo uso indevido da mesma, existem alguns sistemas de irrigação que aproveitam as águas residuais e pluviais. Podemos destacar o do Parque Hermanos Marista em Valência, que aproveita as águas residuais de toda a cidade para regar o parque, que tem uma extensão de 20 mil metros quadrados. Mais recentemente, podemos destacar um caso em Portugal, na Figueira da Foz, no qual as águas residuais são tratadas numa estação de tratamento, abastecendo, de seguida, auto-tanques da Câmara Municipal. Estas águas são utilizados na rega de espaços verdes, árvores e lavagem de ruas da cidade. A utilização de águas residuais requer no entanto cuidados uma vez que, se forem mal utilizadas, podem tornar-se uma fonte de doenças. A utilização de águas pluviais para sistemas de irrigação é, assim, mais simples, pois estas não necessitam de tratamento, sendo apenas necessário proceder ao seu armazenamento em reservatórios [6].

1.2 Motivação

A água é um bem cada vez mais escasso a nível mundial uma vez que existe apenas 3% de água doce em todo o planeta. Assim, a preocupação com este recurso tem vindo a aumentar cada vez mais. Tal facto, acrescido ao contínuo aumento populacional, veio exigir novas técnicas de agricultura sendo uma delas a irrigação. Esta técnica é uma ferramenta imprescindível para a conjuntura económica actual, na qual o mercado é cada vez mais exigente pois existe uma concorrência feroz, tornando-se então necessário a existência de espírito inovador e, principalmente, de eficiência na gestão de recursos. O mercado agrícola embora muito abrangente não é diferente. É então necessário produzir mais, com menos matéria prima.[7]

De modo geral, ao iniciar um projecto de irrigação deve-se ter em mente o desenvolvimento de sistemas de irrigação eficientes que permitam economizar as despesas, a água e o trabalho.

1.3 Objectivos

Partindo do princípio que as variações climáticas ocorrem de forma gradual, não havendo alterações bruscas num curto espaço de tempo, pode ser feito o controlo de um sistema de irrigação utilizando um barramento *1-Wire* e linguagem Java apesar do barramento *1-Wire* ser lento e a linguagem Java não ter garantias temporais.

1.4 Pressupostos

Este trabalho tem como objectivo o desenvolvimento de um sistema de aquisição, monitorização e controlo para aplicações meteorológicas e de engenharia agrícola. O desenvolvimento deste sistema baseia-se na utilização conjunta de um PC, uma plataforma TINI, sensores *1-Wire* e linguagem Java. No final, pretende-se obter um conjunto de sensores e actuadores e uma pequena estação de controlo de rega com acesso remoto.

1.5 Estrutura da Dissertação

No segundo capítulo é feita uma breve descrição dos métodos de irrigação, do funcionamento dos controladores de irrigação e apresentada uma visão geral dos controladores existentes no mercado.

No terceiro capítulo é efectuada uma análise dos sensores e actuadores utilizados

na irrigação sendo dada maior relevância aos sensores de humidade do solo e de pluviosidade.

No quarto capítulo é feita uma introdução a alguns conceitos fundamentais para o desenvolvimento deste projecto. Mais concretamente sobre Java, TINi e *1-Wire*.

No quinto capítulo é feita uma breve descrição do controlo difuso e do modo como este pode ser utilizado em sistemas embutidos.

No sexto capítulo é efectuada uma descrição do *hardware* e *software* implementado.

No sétimo capítulo são descritos alguns testes e conclusões.

No oitavo capítulo são apresentadas as conclusões deste trabalho e são sugeridos trabalhos futuros.

Capítulo 2

Irrigação

A irrigação é uma técnica agrícola que tem por finalidade o fornecimento controlado de água para as plantas, em quantidade suficiente e no momento exacto, assegurando a produtividade e a sobrevivência das mesmas. A irrigação colmata a falta de precipitação natural e, geralmente, enriquece o solo com a deposição de elementos fertilizantes que podem beneficiar a cultura. Torna-se, por isso, necessário um bom programa de irrigação.

A determinação da quantidade de água necessária para a irrigação é um dos principais parâmetros para um correcto planeamento, dimensionamento e manejo de qualquer sistema de irrigação [6]. O solo, o clima e as plantas são outros factores a ter em conta no planeamento dum sistema de irrigação.

2.1 Interacções da irrigação com o solo, plantas, clima e água

Os princípios básicos da irrigação são: fornecer água às plantas de modo a permitir o seu normal desenvolvimento e produção; promover a lixiviação e diluição do excesso de sais em locais que tenham problemas de salinidade [8].

Para análise desses dois pontos há que ter em conta o solo, as plantas, o clima, a água e o sistema de irrigação.

2.1.1 Solo

Para uma boa irrigação é necessário que se conheçam algumas características do solo para que a aplicação de água seja feita de maneira racional, podendo assim evitar problemas como erosão e gastos desnecessários de água e energia. As principais ca-

racterísticas que devem ser analisadas são a capacidade de retenção e a velocidade de infiltração de água no solo [9].

2.1.2 Plantas

Para que o sistema de irrigação seja funcional é indispensável conhecer as plantas: a espécie, a fase de desenvolvimento e o espaçamento das culturas. Torna-se necessário fazer medições ou observações para definir a existência de deficiências de água, tais como o encurtamento de entrenós, o enrolamento e o ângulo das folhas. Este método tem o inconveniente de estas deficiências serem detectadas tardiamente reduzindo assim a possibilidade de recuperação das plantas.

2.1.3 Clima

Neste campo é necessário ter em conta os seguintes factores: precipitação, humidade relativa, temperatura, velocidade do vento e evapotranspiração. A evapotranspiração corresponde à soma dos componentes de transpiração e evaporação. A definição de evapotranspiração é de grande importância uma vez que estabelece o consumo de água pelas plantas e, por conseguinte, a quantidade de água a ser utilizada no sistema de irrigação.

2.1.4 Água

É indispensável conhecer a disponibilidade e a qualidade da água. Este conhecimento é essencial para o sucesso da utilização dos sistemas de irrigação sendo no entanto, a sua qualidade muitas vezes negligenciada. Deste modo, a irrigação poderá produzir efeitos indesejáveis na condução de uma cultura e servir de veículo para a contaminação da população por ingerirem alimentos por ela contaminados. Estes conhecimentos são indispensáveis para determinar o método de irrigação a utilizar [10].

2.2 Métodos de Irrigação

Os principais métodos de irrigação são os seguintes:

- Superfície ou Gravidade;
- Localizada;
- Aspersão.

2.2.1 Superfície ou Gravidade

Este sistema está normalmente associado à agricultura e é o método mais antigo. Este é o processo de irrigação em que a água é conduzida e distribuída sobre a superfície do solo por gravidade. Geralmente é necessária a preparação do terreno para que este apresente condições de condução e distribuição da água. O facto de haver perdas significativas com este tipo de rega, (cerca de 40%) faz com que este sistema seja cada vez menos utilizado [10].

Existem três sistemas de irrigação à superfície: rega por sulco; por faixa e por inundação.

2.2.1.1 Rega por sulco

É um método de irrigação em que a água é aplicada em pequenos canais ou sulcos colocados paralelamente à fileira das plantas e adapta-se à maioria das culturas tais como: a batata, o feijão, o algodão e o milho. [10].



Figura 2.1: Rega por sulco

2.2.1.2 Rega por faixa

Neste sistema, a água é aplicada em faixas de terra com pouco declive longitudinal, separadas por pequenas elevações chamadas diques ou taipas e é muito utilizado na irrigação de pastagens.



Figura 2.2: Rega por faixa

2.2.1.3 Rega por inundação

Neste método de irrigação a água é aplicada em bacias ou tabuleiros de modo intermitente ou permanente. É utilizado no cultivo do arroz, do algodão e em pastagens. É um dos métodos mais usado no mundo.



Figura 2.3: Rega por inundação

2.2.2 Localizada

Este é um sistema de rega que só fornece água às zonas de terreno pretendidas. A água é distribuída através de dispositivos apropriados (emissores) instalados nas tubagens em intervalos regulares de distribuição.

Existem fundamentalmente dois tipos de rega localizada: Rega gota-a-gota e por microaspersão.

2.2.2.1 Rega gota-a-gota

Na rega gota-a-gota a água é distribuída em vários pontos do terreno através de gotejadores. Neste sistema a água é levada sob pressão por tubos, até ser aplicada ao solo

através de emissores directamente sobre a zona da raiz da planta, em alta frequência e baixa intensidade. É utilizado maioritariamente em culturas perenes e em fruticultura mas é também usado por alguns produtores de hortaliças e flores. Há uma tendência do mercado para a utilização cada vez maior do método de rega com o sistema gota-a-gota enterrada pois, é considerada de alta eficiência, na ordem dos 90% [6]. A decisão da utilização do sistema enterrado deve ser tomada depois de analisada criteriosamente a cultura a ser irrigada.



Figura 2.4: Gotejamento

2.2.2.2 Rega por microaspersão

Na rega por microaspersão a água é fornecida a pequenas superfícies por meio de pequenos aspersores ou pulverizadores que lançam jactos de água ao ar e caem em forma de chuva. Existem sistemas móveis e fixos. Este tipo de rega é muito utilizado em jardins. Normalmente são utilizados microaspersores de rotação que podem ter vários ângulos [6].



Figura 2.5: Microaspersores

Em jardins e espaços verdes são utilizados três tipos de implantações de microaspersores: em quadrado, em rectângulo e em triângulo.

A disposição em quadrado é a mais comum.

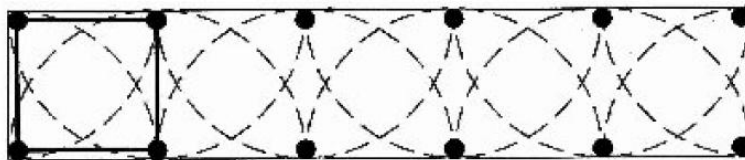


Figura 2.6: Disposição em quadrado

A disposição em rectângulo também é bastante usual pois, muitas vezes, os canteiros têm esta forma. Este posicionamento reduz o efeito adverso do vento.

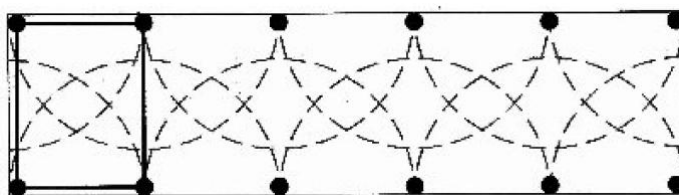


Figura 2.7: Disposição em rectângulo

A disposição em triângulo, é considerada por muitos especialistas como a melhor forma de distribuir os microaspersores, pois esta proporciona uma melhor distribuição da água.

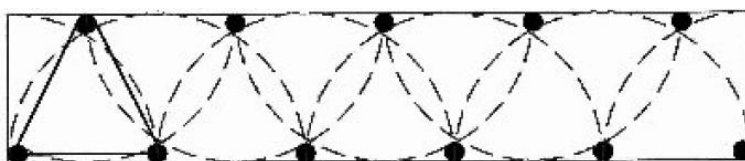


Figura 2.8: Disposição em triângulo

Normalmente em espaços verdes e jardins, são utilizados microaspersores “Pop-up”. Estes emissores são enterrados e quando começa a rega, saem tendo um comportamento igual aos outros microaspersores. Este tipo de microaspersores tem duas vantagens: diminuição dos actos de vandalismo e melhoramento da estética.

2.2.3 Aspersão

Os sistemas de aspersão são utilizados principalmente em grandes áreas de terreno. Têm uma eficiência de 70% caso não haja condições climáticas adversas, como o vento que afecta a uniformidade da distribuição de água [10]. Este sistema de irrigação é muito parecido com o de microaspersão pois o seu princípio básico é o mesmo. A água é bombeada sob pressão por tubos até terminações com pequenos orifícios achatados sendo então espalhada nas plantas em forma de pequenas gotas. É um

método muito utilizado actualmente, podendo ser fixo ou móvel, com movimentação manual ou mecânica.

Neste sistema destacam-se: o canhão hidráulico e o pivô suspenso que pode ser fixo ou móvel.

2.2.3.1 Canhão hidráulico

O canhão hidráulico dispara água pelo ar sobre o campo. Um grande canhão hidráulico é capaz de cobrir muitos hectares de terreno sem ser movido e é muito utilizado em diversas lavouras como a cana-de-açúcar.



Figura 2.9: Canhão Hidráulico

2.2.3.2 Pivô

O pivô é constituído por uma tubulação metálica que se apoia em várias torres triangulares. Nela estão instalados aspersores promovendo a irrigação uniforme numa área de superfície circular. O pivô suspenso móvel espalha a água deslocando-se por toda a extensão da plantação.



Figura 2.10: Pivô

2.3 Sistemas de controlo de Irrigação

Existem dois tipos de controladores de irrigação:

- Controladores de irrigação por horas;
- Controladores de irrigação baseados em factores meteorológicos.

2.3.1 Controladores de irrigação por horas

Os controladores de irrigação por horas utilizam sistemas de controlo em malha aberta. Estes sistemas de controlo não têm qualquer tipo de realimentação, ou seja, o controlador não obtém informação do objecto a controlar.



Figura 2.11: Controlo em malha aberta

Neste tipo de controlador de irrigação o utilizador define a hora de início de irrigação e o tempo de funcionamento, ou então a hora de início e a hora de fim. Na maioria destes controladores também é possível configurar a periodicidade da irrigação ou a quantidade de água a ser usada por período de irrigação. Embora baratos, estes controladores são pouco eficientes pois não verificam se a quantidade de água introduzida

foi a necessária, uma vez que não são sensíveis a alterações climáticas que influenciam em muito a quantidade de água necessária para as culturas. Segundo Faria & Costa [8] “embora possa ser prático no sentido de programação das operações, este método acarreta deficits e excessos de água, uma vez que as condições climáticas são variáveis ano a ano.”. Um exemplo da falta de eficiência, verifica-se com a pluviosidade, pois o controlador continuará a regar independentemente de ter chovido ou não. Existe assim um desperdício de água e corre-se o risco de danificar a plantação por esta ficar alagada [8].

2.3.2 Controladores de irrigação baseados em factores meteorológicos

Os controladores de irrigação baseados em factores meteorológicos utilizam sistemas de controlo em malha fechada. Estes sistemas de controlo funcionam com realimentação, ou seja, o controlador obtém informação do objecto a controlar.

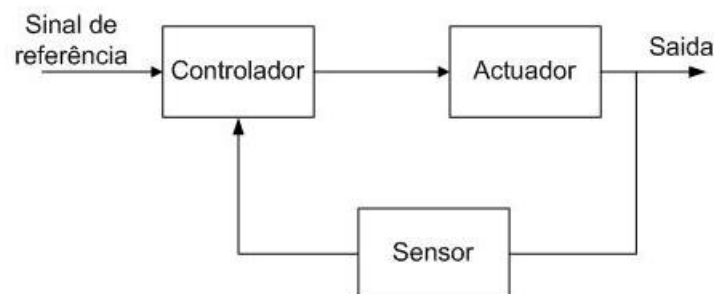


Figura 2.12: Controlo de malha fechada

Este tipo de controladores de irrigação utiliza informação de sensores para determinar quando e que quantidade de água devem aplicar. Os sensores normalmente utilizados na irrigação são os seguintes: Sensor de Humidade do ar, Sensor de Humidade do solo, Sensor de Velocidade do vento, Sensor de Temperatura e Sensor de Radiação solar. Caso não sejam completamente autónomos estes controladores podem necessitar de informação do utilizador tal como os controladores por horas. Embora sejam mais caros e de maior complexidade, este tipo de controladores são muito eficientes uma vez que irrigam a cultura apenas quando necessário e com a quantidade de água necessária [11]. Tomemos como exemplo um controlador semi-autónomo com um sensor de pluviosidade no qual o utilizador define uma hora inicial e o tempo de irrigação. Este controlador mesmo programado para uma determinada hora só irá regar a cultura se o sensor de pluviosidade não detectar que está a chover ou choveu

recentemente.

Tem havido uma grande relutância por parte de pequenas e médias empresas à adesão deste tipo de controladores um vez que requerem um investimento inicial muito elevado. No entanto, estes controladores serão um bom investimento a médio/longo prazo se considerarmos que o investimento inicial pode ser recuperado com a poupança de água e a qualidade das culturas.

2.4 Soluções Comerciais

No mercado existem inúmeros controladores de irrigação por horas e controladores baseados em factores meteorológicos. Os controladores por horas são os mais baratos e mais comuns. Este tipo de controladores costumam ter vários espaços temporais nos quais podem ser definidas horas e dias da semana em que o sistema de irrigação deve funcionar. Alguns destes controladores também têm espaços temporais maiores em que é possível definir o tempo de rega consoante a estação do ano.

Os controladores baseados em factores meteorológicos são também conhecidos como controladores inteligentes. São assim designados pois têm a capacidade de agendar o início e a duração da irrigação. A maioria dos controladores inteligentes existentes no mercado calcula o agendamento através do cálculo da evapotranspiração que é baseada nos dados fornecidos pelos sensores, ou seja, baseiam-se em factores meteorológicos para fazer o agendamento. Hoje em dia existem no mercado dois tipos de controladores inteligentes: de tempo real e baseados no histórico.

Os controladores baseados em tempo real vão fazendo leituras dos sensores e calculando a evapotranspiração baseando-se nos dados actuais. Um exemplo de um controlador deste tipo pode ser um controlador baseado na radiação solar pois, quando a radiação actual é muito elevada este controlador agenda a irrigação para mais tarde.

Os controladores baseados no histórico, vão fazendo leituras dos sensores guardando essa informação e assim, a evapotranspiração será calculada não apenas nas medidas actuais, mas sim com base na informação que vai sendo guardada ao longo do tempo. Um exemplo de um controlador deste tipo pode ser um controlador que guarda a informação da pluviosidade dos últimos dez dias e que calcula quando e quanto deve regar baseando-se não só na quantidade de chuva actual mas também na que caiu nos últimos dias.

Alguns controladores comerciais acabam por ser uma mistura dos dois tipos, por exemplo, um controlador que calcula a evapotranspiração baseando-se nos dados actuais e em dados que são introduzidos na programação sobre estudos meteorológicos

realizados na região.

Hoje em dia existem controladores inteligentes de várias marcas como por exemplo:

- Accurate WheatherSet, baseado em sensores de radiação solar;
- AccuWater, ET Water Systems, baseados em sensores de pluviosidade;
- Alex-Tronix, Aqua Conserve baseados em sensores de temperatura;
- Calense, baseado em sensores ópticos e de pluviosidade;
- Cyber-Rain, baseado em sensores de humidade do ar, pluviosidade e humidade do solo;
- ECO Research, baseado em sensores de radiação solar e temperatura;
- Hunter, baseados em sensores de radiação solar, temperatura e humidade ar;
- Rain Bird, baseado em estações meteorológicas com sensores de humidade do ar, radiação solar, pluviosidade, temperatura e velocidade do vento;
- Entre muitas outras HydroPoint, Hydrosaver, Irrisoft, Irritrol, Tucor, Water2Save.

Como referido anteriormente, estes controladores têm um custo elevado e as soluções comerciais apresentam algumas limitações. A maioria deles tem um número limitado de portos de entrada para sensores. Assim, o preço das estações com menor número de portos (6 a 8) ronda de \$260 a \$1,300 e com mais portos (48 a 66) varia entre \$1,000 e \$6,000, dependendo da marca e do tipo de controlador. Para além disso, a maioria destes sistemas pode conter um módulo de acesso remoto, no entanto, este traz um custo adicional que ronda os \$1,000 [12].

Capítulo 3

Sensores e Actuadores

Um sensor é um dispositivo que recebe um estímulo ou um sinal externo e responde a este. Os sensores foram desenvolvidos com o intuito de realizar medições de alterações físicas podendo ser de medida directa como um termómetro de mercúrio, ou indirecta como um termopar. Os de medida indirecta também podem ser designados por transdutores pois convertem um tipo de energia noutra. Os sensores tornaram-se assim parte do nosso dia a dia, com aplicações em várias áreas como a medicina, a indústria automóvel, a robótica e a agricultura [13].

Existem vários tipos de sensores que podem ser utilizados na agricultura e nos sistemas de irrigação, uma vez que os cultivos são afectados por inúmeros factores físicos como a humidade do solo, a humidade do ar, a temperatura do ar, a temperatura do solo, o pH do solo e a radiação solar. Hoje em dia existem sensores para todos estes factores. No entanto, neste trabalho serão focados com maior relevância os sensores de humidade do solo e de pluviosidade, pois estes são específicos de uma aplicação agrícola o que não é o caso de outros sensores. Enquanto os sensores fornecem informações de entrada para o sistema a partir do mundo externo, os actuadores executam acções de saída. Os actuadores tal como o nome indica são dispositivos que actuam num sistema podendo ser de vários tipos e formas, como servomecanismos ou *lasers*. Nos sistemas de irrigação, os actuadores são vulgarmente electroválvulas.

3.1 Sensores de Humidade do Solo

Existem vários sensores para a medição da humidade do solo. Os mais comuns na agricultura são os seguintes: *Resistivos*, *Capacitivos*, *Tensiómetros*, *Sonda de Neutrões*, *TDR* (*Reflectometria no Domínio do Tempo*) e *DPHP* (*Dual-Probe Heat-Pulse*) [8].

3.1.1 Resistivos

Os sensores de humidade do tipo resistivo são baseados na variação da resistência eléctrica entre dois eléctrodos inseridos num bloco de gesso.

Quando enterrados no solo os blocos de gesso absorvem ou perdem água até que esta esteja em equilíbrio no interior do bloco e no solo. Com a variação da humidade, a resistência entre os eléctrodos também varia mas numa proporção inversa. Para estes sensores, algumas marcas disponibilizam uma curva de calibração do tipo (ohm-KPa), no entanto, esta pode sofrer pequenas variações com a temperatura.

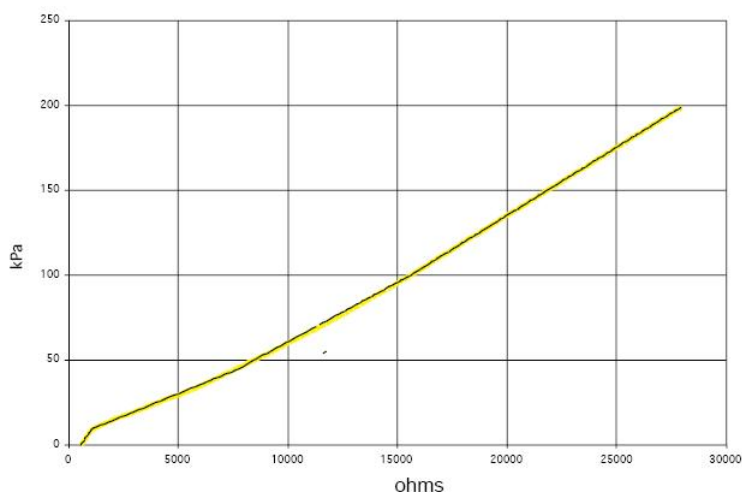


Figura 3.1: Curva de calibração Watermark (Humidade do solo em KPa e Resistência em ohms)

A humidade do solo pode ser obtida indirectamente, através da medição da resistência do bloco de gesso. Para efectuar a leitura da resistência o sensor deve ser alimentado com uma fonte AC por forma a não ficar polarizado.

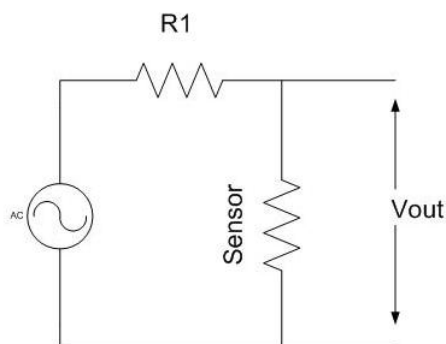


Figura 3.2: Circuito de sensor resistivo

$$V_{out} = \frac{R_{Sensor}}{R_{Sensor} + R_1} \times V_{ac} \quad (3.1)$$

Estes sensores devem ser colocados próximos da raiz e em duas profundidades, por forma a obter uma maior percepção e controlo da água absorvida pela raiz da planta. Este tipo de sensor é muito simples e barato mas omite o efeito da temperatura e da salinidade do solo que afectam a sua leitura e têm um tempo de vida que ronda apenas dois anos [14]. Existem vários tipos de sensores resistivos no mercado sendo o mais comercializado o Watermark.

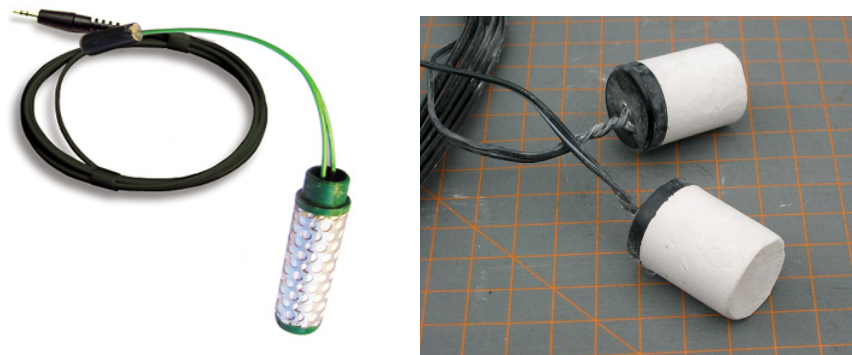


Figura 3.3: Sensores resistivos de humidade do solo

3.1.2 Capacitivos

Os sensores capacitivos são conhecidos por serem utilizados na medição do movimento angular e linear mas também podem ser utilizados na medição de pressão, nível e humidade. Tal como um condensador o sensor capacitivo é muito sensível a dois tipos de parâmetros: a distância entre os eléctrodos e a alteração do dieléctrico. Para a medição da humidade do solo a solução passa por variar o dieléctrico, assim, a capacidade depende apenas da matéria entre eléctrodos. O sensor capacitivo de humidade do solo é constituído por um par de eléctrodos normalmente dispostos de forma paralela ou em anéis de metal, formando assim um condensador onde o solo actua como dieléctrico. Consoante a variação da humidade do solo o dieléctrico varia fazendo com que o sensor apresente uma capacidade diferente nos seus terminais. Desta forma, o sensor pode ser utilizado num circuito oscilador.

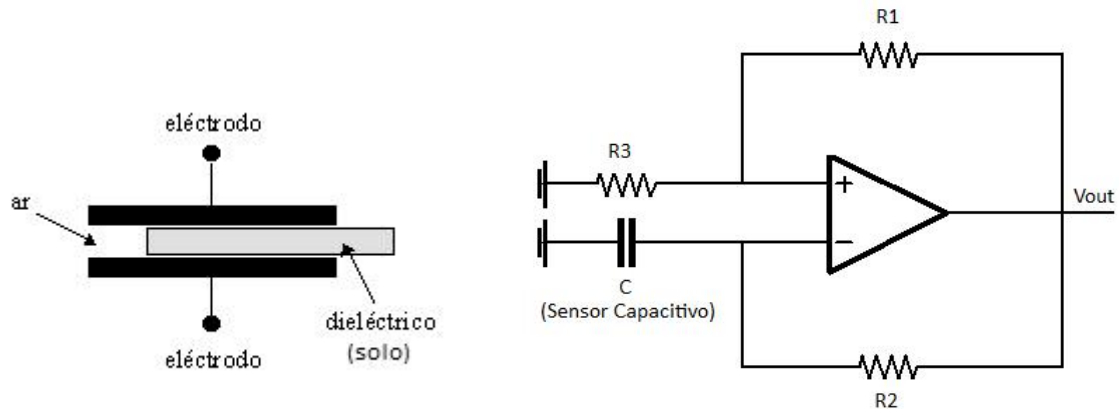


Figura 3.4: Sensor capacitivo e Circuito oscilador

$$f_{Vout} = \frac{1}{2\pi R_2 C} \quad (3.2)$$

As alterações da quantidade de água no solo provocam uma variação de frequência do circuito. Os sensores capacitivos, tal como os sensores resistivos, também são sensíveis ao tipo de solo e à temperatura. Estes sensores devem ser colocados perto da raiz das plantas para se obterem leituras com maior eficiência [15] [16]. A figura 3.5 apresenta uma fotografia de um sensor capacitivo da Deacon Devices [17].

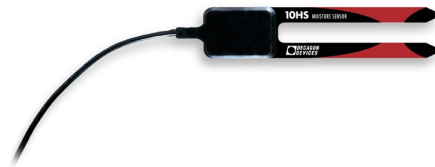


Figura 3.5: Sensor Capacitivo

3.1.3 Tensiómetros

Os tensiómetros são sensores baseados na força exercida pelas plantas para absorver a água existente no solo. Estes sensores são constituídos por um tubo sólido com uma extremidade porosa no qual é inserida água, sendo depois fechado com um medidor de vácuo. Tal como os sensores resistivos, a água no interior do tubo tenta manter o equilíbrio com a do solo, assim, quando a planta absorve água provoca uma diferença de pressão e esta sai do tubo. Como este se encontra pressurizado, o medidor de vácuo mede a diferença de pressão. Desta forma, estes sensores têm uma leitura da quantidade de água no solo. Os medidores de vácuo podem conter dois dispositivos adicionais: um

interruptor que muda de estado quando a pressão passa um determinado nível ou um conversor analógico digital ligado a um sensor de pressão, tornando possível a leitura do medidor de vácuo por circuitos electrónicos. Os tensiómetros são sensores com grande durabilidade uma vez que, se manuseados com cuidado podem durar muitos anos. Devido à sua forma de funcionamento é necessária alguma manutenção pois como a água vai saindo dos tubos estes podem ficar com pouca água efectuando leituras erróneas. Estes sensores tal como os resistivos devem ser colocados perto das raízes das plantas e a duas profundidades [14] [18] [19] [20]. A figura 3.6 apresenta uma fotografia de um tensiómetro da ICT Internacional [21].

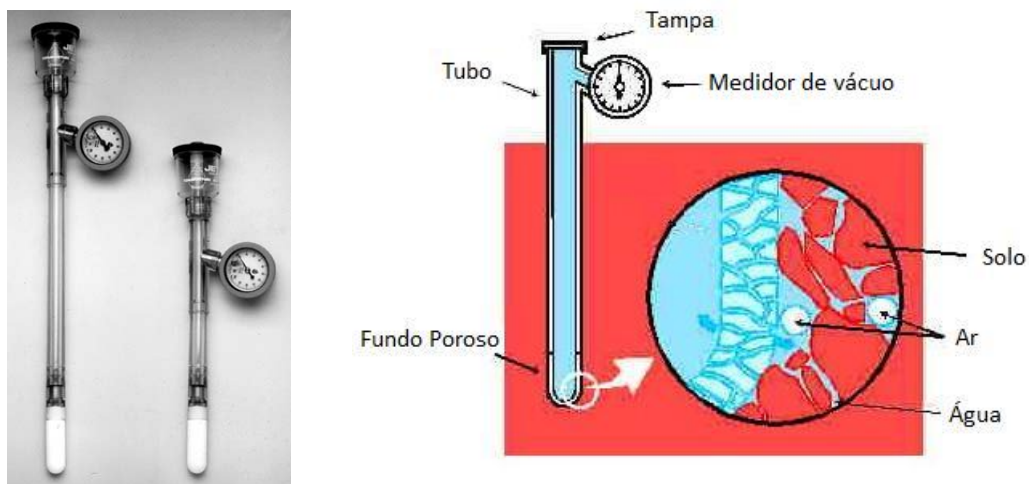


Figura 3.6: Tensiómetros

3.1.4 Sonda de Neutrões

O funcionamento da sonda de neutrões consiste no envio de um fluxo de neutrões que reage com os átomos de hidrogénio existentes na água, permitindo assim, fazer uma contagem do número de partículas de água no solo e calcular, através disso, a humidade do mesmo. Embora sejam dos sensores mas sofisticados e que proporcionam melhores resultados estes têm alguns inconvenientes, pois são extremamente caros, necessitam de uma licença especial e de algum treino para se poderem utilizar [14]. Na figura 3.7 pode-se observar a fotografia de uma sonda de neutrões da Irricampo [22].



Figura 3.7: Sonda de Neutrões

3.1.5 TDR (Reflectometria no Domínio do Tempo)

O TDR é o sensor com tecnologia mais recente na medição da humidade do solo. Este sensor baseia-se nas propriedades das ondas electromagnéticas, na sua propagação e reflexão. O TDR é um emissor electrónico ligado ao solo através de duas hastes. Este dispositivo emite um sinal eléctrico numa das hastes e consoante o grau de humidade do solo este recebe o sinal propagado e reflectido. Estes sensores têm a vantagem da medida ser independente do solo, da temperatura ou da salinidade, no entanto, este tipo de sensores é muito caro e por isso a sua aplicação é ainda muito limitada [14] [16]. A figura 3.8 apresenta uma fotografia de um sensor TDR da Mesa Systems Co [23].



Figura 3.8: Sensor TDR

3.1.6 DPHP (*Dual-Probe Heat-Pulse*)

Este sensor trabalha sobre o mesmo princípio que o TDR mas o DPHP é muito menor, de menor custo e com capacidade de ligação a um barramento que permite uma maior mobilidade. O DPHP é um sensor que contém duas agulhas sendo que uma delas envia um impulso de calor e a outra mede a temperatura determinando a capacidade volumétrica do solo e consequentemente a sua quantidade de água [16]. A figura 3.9 apresenta um sensor DPHP desenvolvido por uma equipa da Universidade do Minho [16].

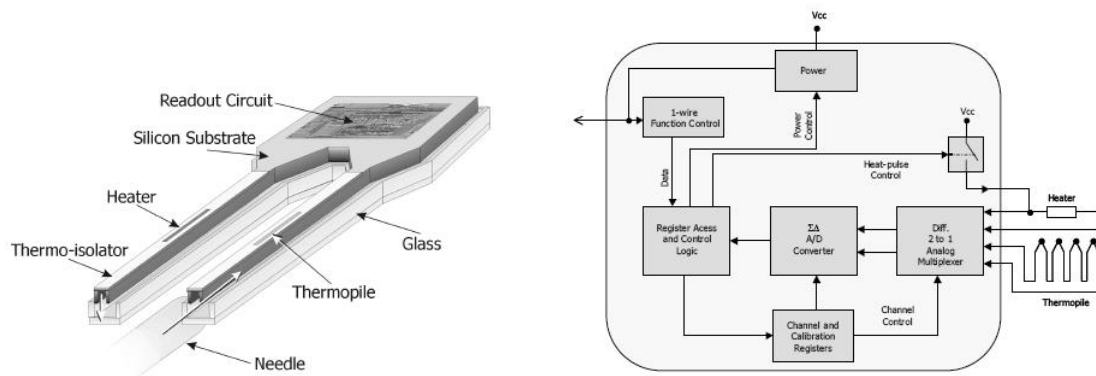


Figura 3.9: DPHP

3.2 Sensores de Pluviosidade

Os sensores de pluviosidade são instrumentos utilizados por meteorologistas para medir a quantidade de líquido precipitado. Hoje em dia, estes são muito comuns em sistemas de irrigação pois permitem evitar regas desnecessárias.

3.2.1 Sensor de chuva

O sensor de chuva é constituído por um copo poroso que tem no seu interior um disco que absorve água. O seu funcionamento é muito simples. Quando chove, o disco absorve água, torna-se mais pesado e activa um interruptor que se encontra no fundo do copo. Este interruptor pode ser utilizado para desligar o circuito de comando das válvulas do sistema de rega. A figura 3.10 apresenta um sensor de chuva da HomeTech Solutions [24].

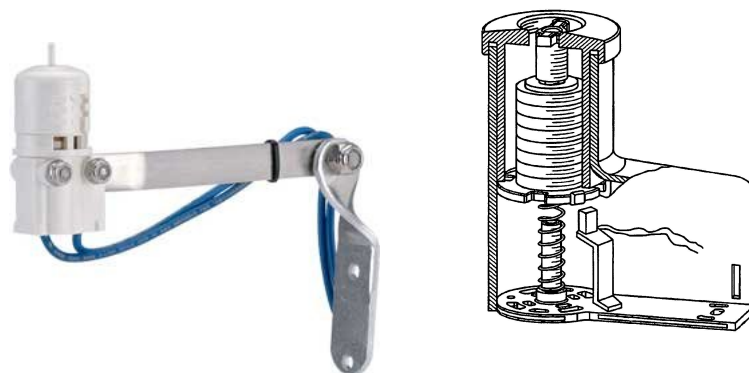


Figura 3.10: Sensor de chuva

3.2.2 Medidor de chuva

O medidor de chuva é constituído por um balde afunilado que permite colectar a água. Esta é encaminhada para um balancé que tem dois pequenos recipientes igualmente divididos. Quando a quantidade de água num dos recipientes é elevada o balancé muda de posição, despejando a água do recipiente cheio e começando a encher o outro recipiente. Assim, para medir a quantidade de água da chuva basta saber a quantidade de água que cada lado do balancé pode conter e o número de vezes que mudou. Para isso é utilizado um *Reed Switch* que permite saber quando o balancé mudou de lado e um contador electrónico para contar o número de vezes que mudou. A figura 3.11 apresenta uma fotografia de um medidor de chuva da MadgeTech [25].

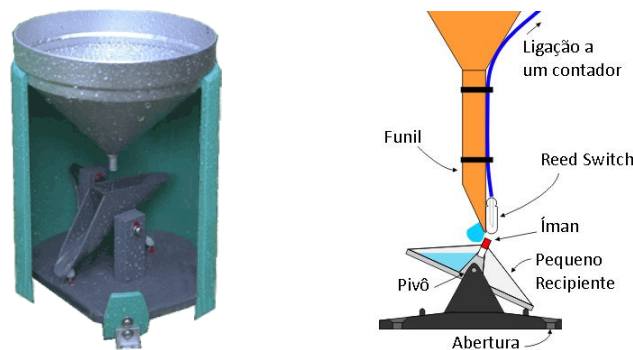


Figura 3.11: Medidor de chuva

3.3 Outros Sensores

Para além dos sensores de humidade do solo e da pluviosidade existem outros sensores que permitem medir outros factores físicos como: a temperatura, a humidade do ar, a radiação solar e a velocidade do vento.

Existem vários tipos de sensores de temperatura: termístors, RTD e termopares. Os termístors medem a temperatura através da variação da resistência baseado em semi-condutores. Os RTD medem a temperatura através da variação da resistência. Os termopares utilizam uma junção entre dois materiais para medir a temperatura, uma vez que, essa junção cria uma diferença de potencial proporcional à temperatura. Na figura 3.12 pode ser observado um sensor de temperatura RTD.

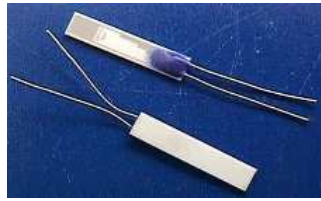


Figura 3.12: Sensor de temperatura RTD

Os sensores de humidade do ar podem ser resistivos ou capacitivos tendo um funcionamento parecido com os sensores da humidade do solo. A figura 3.13 apresenta dois sensores de humidade do ar.



Figura 3.13: Sensor de humidade do ar

A radiação solar pode ser medida através de elementos resistivos que variam com a quantidade de radiação ou com painéis solares que fornecem uma corrente proporcional à quantidade de radiação. Na figura 3.14 pode ser visualizada a fotografia de um painel solar.

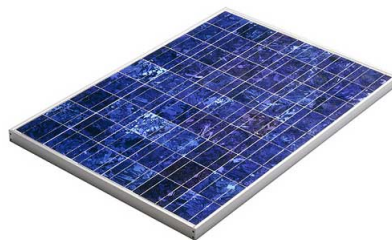


Figura 3.14: Painel solar

Uma maneira simples de medir a velocidade do vento é contar o número de voltas dadas por um tubo com pás numa das extremidades, podendo ser utilizados sensores de posição, interruptores ou *encoders* como pode ser visto na figura 3.15.



Figura 3.15: Velocidade do vento

Nalguns casos em vez de se utilizar um ou outro sensor são utilizadas estações meteorológicas que proporcionam uma informação de todos ou quase todos os factores ambientais, no entanto, este sistema de medição torna-se mais caro. Pode ser observada uma fotografia de uma estação meteorológica na figura 3.16.



Figura 3.16: Estação meteorológica

3.4 Electroválvulas

As electroválvulas funcionam como actuadores num sistema de irrigação. Estas são formadas por duas partes: circuito magnético e corpo. O circuito magnético é constituído por diversas peças: bobina, tubo, culatra, anel de desfasagem e molas. O corpo inclui orifícios de ligação, membrana, pistão, e muitas outras peças que dependem da tecnologia utilizada na electroválvula. A maioria das electroválvulas funcionam com sistemas *on/off*, ou seja, a válvula está completamente aberta ou completamente fechada. Existe ainda outro tipo de electroválvulas denominadas proporcionais podendo ter valores intermédios de caudal de água, no entanto, normalmente estas não são utilizadas na irrigação [26].

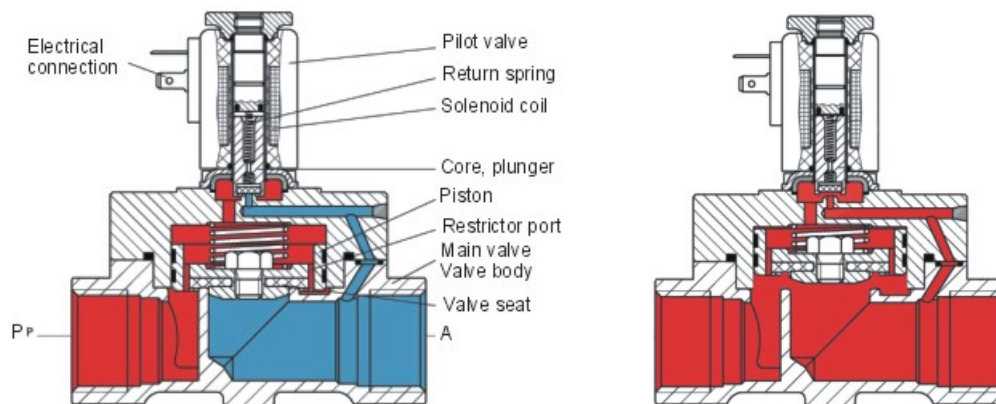


Figura 3.17: Electroválvulas

Nos sistemas de irrigação são utilizadas electroválvulas *on/off*, existindo três tipos destas electroválvulas:

- **Electroválvulas AC** são alimentadas com corrente alterna. A abertura e o fecho das electroválvulas dependem da posição do núcleo móvel que é deslocado com o campo magnético produzido pela bobine quando excitada. Quando não passa corrente na bobine a válvula fica fechada e quando se aplica uma corrente a bobine cria um campo que puxa um êmbolo móvel abrindo a válvula. Este tipo de electroválvulas é usualmente alimentado a 24V;
- **Electroválvulas DC** são alimentadas com corrente contínua. Este tipo de electroválvulas tem um funcionamento igual ao das electroválvulas AC sendo também alimentadas normalmente a 24V [27];
- **Electroválvulas por pulso** mudam de estado quando sujeitas a pulsos. A abertura e fecho deste tipo de válvulas está dependente da direcção do pulso, tendo uma relação inversa como pode ser visto na figura 3.18. Estas electroválvulas contêm um íman permanente com um campo magnético fraco, quando o pulso é aplicado aumenta fortemente o campo do lado do íman permanente, abrindo a válvula e quando o pulso acaba, o campo criado pelo solenoíde desaparece, mas o êmbolo fica seguro ao íman permanente. Para fechar a válvula aplica-se um pulso no sentido contrário, criando um campo maior do lado oposto ao íman permanente, puxando o êmbolo para baixo. Este tipo de válvulas permite poupar energia, uma vez que esta apenas é consumida quando a válvula abre ou fecha. Tal facto permite que estas electroválvulas sejam alimentadas a pilhas [28].

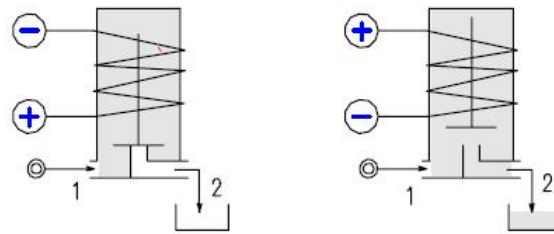


Figura 3.18: Electroválvula por pulso [1]

Capítulo 4

Sistemas Embutidos Baseados em Linguagem Java

A utilização de linguagem Java em sistemas embutidos é uma área com grande actividade. São exemplo disso os telemóveis. Outro exemplo possível é o sistema TINi da *Dallas Maxim*. Esta plataforma dotada de uma Máquina Virtual Java possui várias interfaces especialmente dedicadas à aquisição e controlo de dados. Uma das interfaces disponíveis é a do barramento *1-Wire*. A *Dallas* disponibiliza uma enorme variedade de componentes capazes de serem directamente ligados a este tipo de barramento possibilitando uma implementação simples e eficiente de sistemas que vão desde a domótica às aplicações agrícolas e de meteorologia.

4.1 Linguagem Java

Java é uma linguagem de programação, originalmente desenvolvida por uma equipa de programadores liderada por James Gosling, da *Sun Microsystems*. Esta linguagem herdou grande parte da sua sintaxe de linguagens como C ou C++ mas com um modelo mais simples, orientado a objectos e com algumas facilidades de camadas de baixo nível criando uma maior abstracção. Criada inicialmente em 1995, com a finalidade de gerar aplicações para páginas WEB, esta linguagem contém poderosas bibliotecas gráficas e de comunicação. Desde então, a Sun tem vindo a desenvolver vários compiladores, Máquinas Virtuais e bibliotecas, sendo que em Maio de 2007, a empresa decidiu tornar a maior parte da sua tecnologia Java como *software* livre.

4.1.1 Máquina Virtual Java

A máquina virtual Java (JVM - *Java Virtual Machine*) é um mecanismo que permite executar código Java em qualquer plataforma. Os programas Java não são compilados directamente para linguagem máquina mas sim para uma representação intermédia denominada *bytecodes*. A Máquina Virtual torna-se assim num tradutor que converte *bytecode* em código nativo. Segundo a definição da *Sun*, a JVM pode ser entendida como "uma máquina imaginária implementada via *software* ou *hardware* que executa instruções vindas de *bytecodes*".

Bytecode é uma codificação que traduz o programa para um formato que a JVM seja capaz de executar. Quando um código em Java é compilado, um arquivo com a extensão *.class* é gerado sendo este o arquivo do tipo *bytecode*. Assim, quando um programa com o nome `MyProgram.java` for compilado, um arquivo chamado `MyProgram.class` é criado podendo este ser executado por uma JVM.

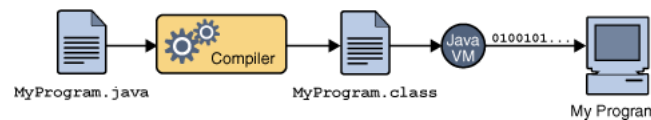


Figura 4.1: Compilar em Java[2]

Como actualmente existem Máquinas Virtuais para a maioria dos sistemas operativos e tendo em conta que os *bytecodes* são os mesmos independentemente do sistema operativo, isto faz com que os programas Java sejam altamente portáteis. Podendo por exemplo um programa Java criado em Windows ser executado em Linux ou em qualquer outro sistema operativo desde que este contenha uma JVM.

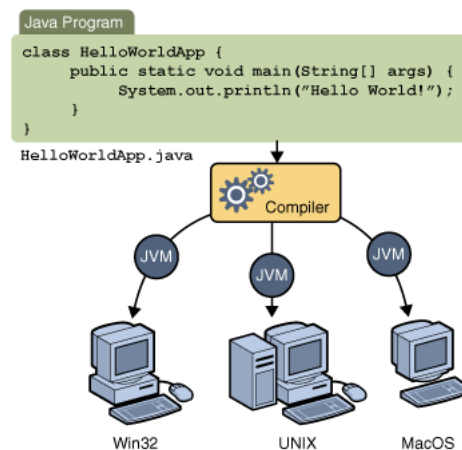


Figura 4.2: Independência do Sistema Operativo em Java [2]

Por ser altamente portátil a comunidade científica considera que o código imple-

mentado em Java tem baixo desempenho. Há, no entanto, quem defenda o contrário, uma vez que existem alguns algoritmos de optimização que minimizam este problema.

Desde o seu início que a linguagem Java se preocupou com a questão da segurança e, como tal, é quase impossível a criação de programas em Java para fins maliciosos. Quando um programa em Java é executado, os *bytecodes* precisam passar pelos requisitos de segurança presentes na JVM, o que impede a execução de programas com irregularidades. Assim, se por exemplo, no programa houver instruções para aceder a áreas restritas de memória ou aceder a recursos de *hardware* sem permissão, a JVM não aprovará o código [29] [30].

4.1.2 Vantagens e desvantagens

A linguagem Java tem as seguintes vantagens:

- **Portabilidade** - Esta é uma das grandes vantagens de Java, pois o código torna-se independente do sistema operativo, ou seja, altamente portátil, no entanto, como Java compila para *bytecode*, e só depois transforma em código nativo, faz com que a maioria das pessoas o considere lento, para além de que, o código nativo produzido pela Máquina Virtual, não se encontra na maioria das vezes optimizado.
- **Gestão de memória** - Como uma linguagem orientada a objectos, Java trabalha quase sempre com referências. Essas referências são alocadas de forma abstracta na memória, o que é bom, pois o programador não tem que se preocupar com a gestão de memória, no entanto, esta abstracção cria um problema, a memória vai sendo sempre alocada, e apenas é libertada quando o sistema chama o *garbage collector*, o que é feito em tempo aleatório. Assim, o *garbage collector* quando chamado, pára a execução de todos os outros processos e verifica quais os locais de memória não referenciados e liberta-os, isto faz com que Java não tenha garantias temporais [31].
- **Simplicidade** - É uma linguagem simples, possuindo baixa complexidade no ciclo de desenvolvimento;
- **Segurança** - Possui um conjunto de regras que não permite a execução de aplicações vindas de fontes desconhecidas;
- **Recursos de Rede** - Possui uma vasta biblioteca de classes que facilitam a cooperação com protocolos TCP/IP, como HTTP e FTP;

- **Bibliotecas** - Java contém um vasto conjunto de bibliotecas (API);
- ***Multithread*** - Possui facilidades na criação de programas distribuídos e multi-tarefa (múltiplas linhas de execução num mesmo programa);

A linguagem Java, em sistemas embutidos, tem as suas desvantagens tais como:

- **Escalabilidade** - Por defeito Java inclui uma larga lista de classes não escaláveis, ou seja, Java contém um conjunto de classes fundamentais que não podem ser aumentadas;
- **Velocidade** - É uma linguagem semi-compilada o que faz com que tenha uma execução mais lenta do que a linguagem máquina;
- **Controlo de *Hardware*** - Não proporciona acesso directo ao *interrupt handling*.

No âmbito deste projecto a utilização de Java torna-se vantajosa. Como o sistema a controlar é bastante lento, faz com que as desvantagens de Java como a lentidão e a não optimização de código, não sejam significativas. É de se salientar que Java tem outra vantagem favorável a este projecto, dado que contém uma biblioteca de *1-Wire*, que facilita em muito a sua utilização.

4.1.3 Alguns Sistemas embutidos baseados em Java

Um sistema embutido é aquele em que o computador é completamente dedicado ao dispositivo que controla. Diferente do computador de propósito geral, um sistema embutido realiza um conjunto de trabalhos predefinidos, geralmente com requisitos próprios [31].

Sendo dedicado a tarefas específicas, através de engenharia pode-se optimizar o projecto diminuindo o tamanho, os recursos computacionais e o custo do produto. Normalmente estes sistemas têm recursos limitados.

Hoje em dia é possível encontrar sistemas embutidos em quase tudo ao nosso redor. De entre inúmeros exemplos encontram-se telemóveis, automóveis, torradeiras, semáforos e até mesmo escovas de dentes. No início, os sistemas embutidos eram desenvolvidos em linguagem máquina mas, com o passar do tempo os fabricantes decidiram utilizar linguagens de mais alto nível como C e C++. Porém, estas linguagens têm um grau de complexidade elevado e por conseguinte, necessitam um maior tempo de desenvolvimento o que implica um custo mais elevado. Por este motivo a linguagem Java torna-se uma solução cada vez mais procurada pelos fabricantes.

Actualmente existem vários sistemas embutidos com microcontroladores baseados em Java, entre os quais podemos destacar JStik, TINI400 TStik2, JStamp, SaJe e SNAP. A tabela 4.1 apresenta uma pequena comparação destes sistemas.

Comparando as características dos vários sistemas e tendo em conta que neste projecto se pretende usar acesso remoto (Ethernet), *1-Wire* e CAN, decidiu-se utilizar a plataforma TINI400 da *Maxim*, uma vez que esta cumpre todos requisitos e mantém uma boa relação qualidade preço.

4.2 Plataforma TINI

Dallas Semiconductor criou a plataforma TINI (*Tiny InterNet Interface*) que tem o objectivo de facilitar o desenvolvimento de aplicações baseadas em Java. Esta plataforma é formada pela combinação de um conjunto de circuitos integrados e um ambiente de programação baseado em Java sendo criada com o intuito de manter a comunicação com actuadores e sensores de um sistema de automação. A ligação de dispositivos de entrada/saída à plataforma TINI torna possível não só um controlo local mas também um controlo global pois esta contém protocolos de redes TCP/IP e um ambiente de programação Java. Esta capacidade de comunicação proporciona uma fácil interacção com os dispositivos associados à plataforma TINI, podendo ser acedidos remotamente.

4.2.1 TINI400

A plataforma TINI400 é constituída por uma placa DSTINIm400 e por uma *motherboard* conhecida como DSTINIs400. A combinação das duas placas cria uma plataforma que permite o desenvolvimento de aplicações.

4.2.1.1 DSTINIm400

O DSTINIm400 é um módulo que implementa a plataforma TINI contendo um microcontrolador DS80C400 desenvolvido pela *Dallas Semiconductor*, baseado no popular 8051 da *Intel*. No entanto, o DS80C400 é mais rápido. A diferença de velocidade deve-se a uma frequência de relógio maior e à capacidade de executar um maior número de instruções por ciclo de relógio. O DS80C400 tem uma frequência de relógio máxima de 75MHz e um ciclo mínimo de cada instrução de 54ns [32] [33]. Este microcontrolador contém:

- Um controlador Ethernet, que suporta ligações de 10/100-Mb/s;
- Três portas série assíncronas;

Nome	TStik2 TINI400	Javelin Stamp	SNAP	JStik	JStamp+	SaJe
JVM edition, type, size	custom 1.1.8 version firmware, 448 KByte	subset of JavaCard	J2ME/CLDC plus java.net	J2ME/CLDC native, 0 KBytes (in silicon)	J2ME/CLDC native, 0 KBytes (in silicon)	J2ME/CLDC native, 0 KBytes (in silicon)
# of threads	16 max	1	?	unlimited (to max heap size)	unlimited (to max heap size)	unlimited (to max heap size)
SRAM	512 KB or 1 MB	32 KBytes	8 MB DRAM	2 MB	512K	1 MB
Flash	2 MBytes	32 KBytes	2 MB	4-8 MB	2 MB	4 MB
UARTS	1x RS232, 2x TTL 1x 1Wire 115 kbaud	three full duplex or 7 one-way in firmware, 57.6 kbaud	three full duplex	2x RS232 or TTL, 115 kbaud	2x TTL, 115 kbaud	1x RS232 1x RS232 or 1Wire, 115 kbaud
javaxcomm serial I/O	yes	no	yes	yes	yes	yes
I/O interface	SPI, I2C, CAN	?	SPI, I2C, CAN	SPI, I2C	SPI, I2C	SPI, I2C
ethernet	10/100BaseT	no	10/100BaseT	10BaseT	no	10BaseT
1Wire net	yes - onboard	no	yes - onboard	yes	yes	yes - onboard
1Wire Java API support	yes	no	yes	yes	yes	yes
Size (inches)	4.25 x 1.25 SIMM72/ SODIMM	1.2 x 0.3 DIP24	4.25 x 1.25 SIMM72	3.00 x 2.65 SIMM30	1.00 x 2.00 DIP40	3.9 x 6.2 Euroboard
Voltage range	5V	5-15V	5V	3.3 or 5-14V	3.3 or 5-14V	6-20 VDC
cost @10	\$99	\$80	\$148	\$299	\$105	\$399

Tabela 4.1: Comparação de sistemas embutidos baseados em Java [4]

- Um controlador CAN 2.0B (*Controller Area Network*). Este é um protocolo série bastante utilizado na indústria automóvel e em aplicações médicas;
- 64 portos I/O de uso genérico.

Esta placa contém também:

- 1MB de memória RAM e 1MB de *flash*;
- Uma ROM de 64 KBytes, programável que contém o *firmware*, mais concretamente a *stack* TCP/IP, protocolos relacionados e um escalonador de tarefas que permite a partilha do CPU por várias tarefas;

O DSTINIm400 é alimentado a 3.3V e o DS80C400 é alimentado a 1.8V. A placa DSTINIm400 interliga-se com o exterior, através de um conector SODIMM de 144 pinos, tal como pode ser observado an figura 4.3.



Figura 4.3: DSTINIm400

4.2.1.2 DSTINIs400

A DSTINIs400 é uma *motherboard* que proporciona ligações físicas para o desenvolvimento de aplicações para a DSTINIm400 [32] [34]. Assim esta placa contém:

- Um *transceiver* Ethernet (LXT972A);
- Dois conectores TTL/RS232;
- Um *transceiver* CAN 2.0B;
- Conversor série/*1-Wire*;
- Um conector de 48 pinos I/O do tipo paralelo;
- Um conector SODIMM de 144 pinos onde é ligado o módulo DSTINIm400.

Esta placa é alimentada com 5V e contém um regulador de tensão de 3.3V para alimentar a placa DSTINI_m400. A figura 4.4 apresenta uma fotografia de uma placa DSTINI_s400.

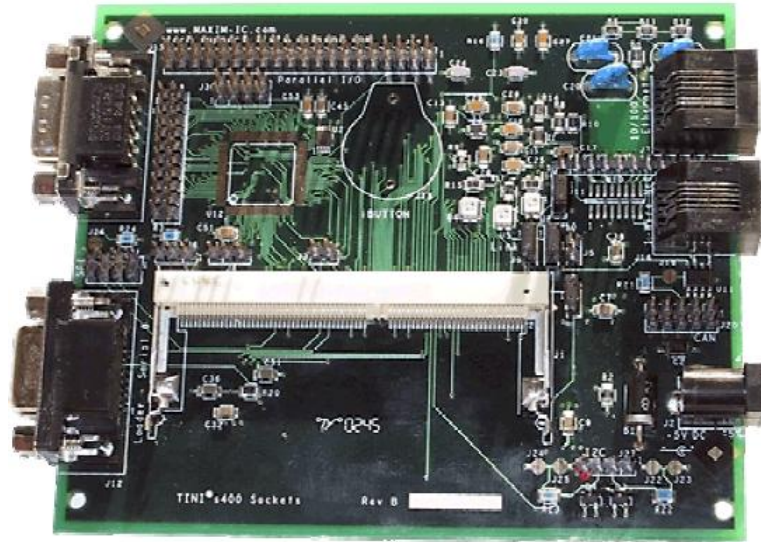


Figura 4.4: DSTINI_s400

4.2.2 Sistema Operativo TINI

O sistema operativo TINI pode ser dividido em três grupos como pode ser observado na figura 4.5 [35]:

- Escalonador de Processos e *threads* - O sistema operativo TINI proporciona um ambiente multi-tarefa, ou seja, escalona as tarefas em diferentes janelas temporais. O escalonamento é do tipo *round-robin* disponibilizando uma janela de execução de 8ms para cada processo. As *threads* são escalonadas da mesma forma mas com janelas de execução de apenas 2ms.
- Gestor de memória - É o responsável por guardar os dados nas zonas de memória específicas e pela execução do GC (*garbage collector*).
- Gestor de I/O - Este é responsável pela gestão dos dispositivos de entrada e saída incluindo CAN e as linhas série.

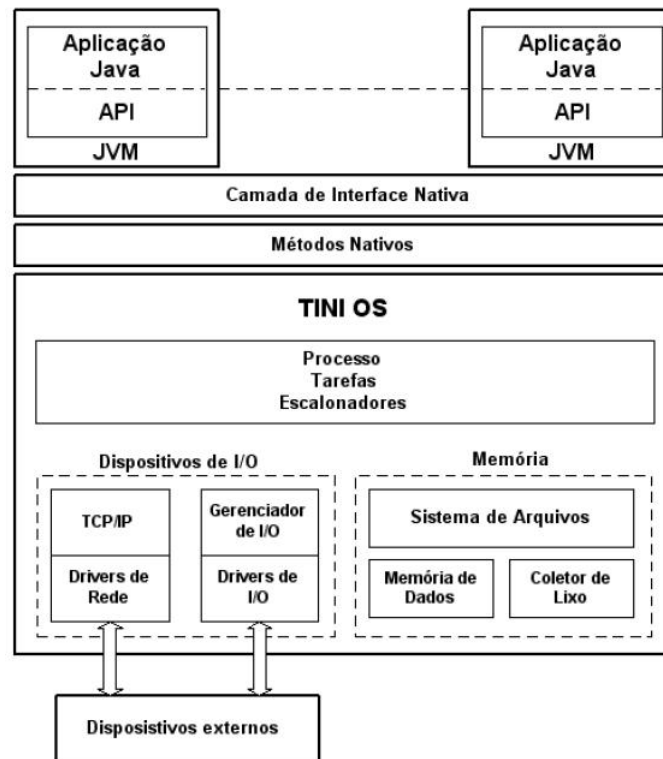


Figura 4.5: TINI OS

4.2.3 Máquina Virtual Java e API

A JVM utilizada pela TINI ocupa apenas 40KBytes, mas inclui uma longa lista de *packages* Java [35].

- *java.lang* - *Package* que contém as classes mais básicas da linguagem de programação Java;
- *java.net* - Classes para implementação de aplicações de redes, como por exemplo implementação de *sockets*;
- *java.io* - Implementação de sistemas de entrada/saída;
- *java.util* - Classes para manipulação de data e hora;
- *javax.comm* - Permite a interação com interfaces paralelas e série;
- *com.dalsemi.comm* - Classes de baixo nível para controladores CAN e portas série;
- *com.dalsemi.fs* - Extensão da classe *java.io.file*;

- *com.dalsemi.io* - Classes de auxílio a conversão de código entre interfaces I/O;
- *com.dalsemi.onewire* - Classes de acesso a dispositivos *1-Wire*;
- *com.dalsemi.protocol* - Classes usadas por diversos protocolos de rede;
- *com.dalsemi.shell* - Contém comandos de implementação e controlo de Protocolos de Transferência de arquivos;
- *com.dalsemi.system* - Classe de acesso ao microcontrolador;
- *com.dalsemi.tininet* - Classes de configuração e controlo de dispositivos de rede.

4.3 Barramento *1-Wire*

O protocolo *1-Wire* foi desenvolvido pela Dallas Semiconductor actualmente Maxim. Na definição habitual, este barramento é constituído por três elementos: um computador que contém um programa de controlo e gestão de actividades seguindo um protocolo de comunicação, dois a três condutores que constituem a ligação física e dispositivos remotos *1-Wire*. O *1-Wire* é um protocolo do tipo *master-slave*. Entende-se como *master* o dispositivo capaz de controlar e gerir a transmissão de dados e como *slave* o dispositivo endereçado e gerido pelo *master*. De acordo com a definição anterior o computador é considerado como *master* e os outros dispositivos tal como sensores são considerados como *slaves* [36]. Para uma maior versatilidade e simplicidade este protocolo permite um único *master* no barramento e utiliza um controlo rígido, não sendo possível qualquer tipo de comunicação entre *slaves*. Todos os dispositivos *1-Wire* possuem um código de identificação de 64 *bits* sendo único para cada dispositivo tornando-se assim fácil a sua identificação no barramento. Este código é gravado a laser numa ROM (*Read-Only-Memory*) que se encontra no interior dos dispositivos, os primeiros 8 *bits* são reservados para identificar a família do dispositivo, os 48 *bits* seguintes são o número de série e os últimos 8 *bits* estão reservados para identificação de erros sendo utilizado um código de redundância cíclica relacionado com os primeiros 56 *bits* [37]. Os dispositivos *1-Wire* contêm também um circuito de protecção contra descargas electrostáticas, o que permite que estes possam entrar em contacto com o corpo humano sem serem destruídos.

4.3.1 Transmissão de dados

No início *1-Wire* tinha sido desenvolvido para pequenas redes com poucos dispositivos e pequenas distâncias ao *master*. Depois, com o tempo e aplicações que exigiam cada

vez maiores distâncias e um número maior de dispositivos, o protocolo foi sendo alterado para suportar estas exigências e proporcionar uma melhor transmissão de dados em situações extremas. A transmissão de dados no protocolo *1-Wire* é feita de forma assíncrona, ou seja, cada pulso digital que compõe o protocolo de comunicação, transmitido em formato binário, é caracterizado por intervalos de tempo distintos [38]. Este tipo de comunicação é possível, pois cada dispositivo *1-Wire* contém um temporizador interno capaz de gerar pulsos digitais predefinidos pelo protocolo de comunicação *1-Wire*. Este temporizador é sincronizado com o *master* quando a linha de dados transita do valor lógico um para zero.

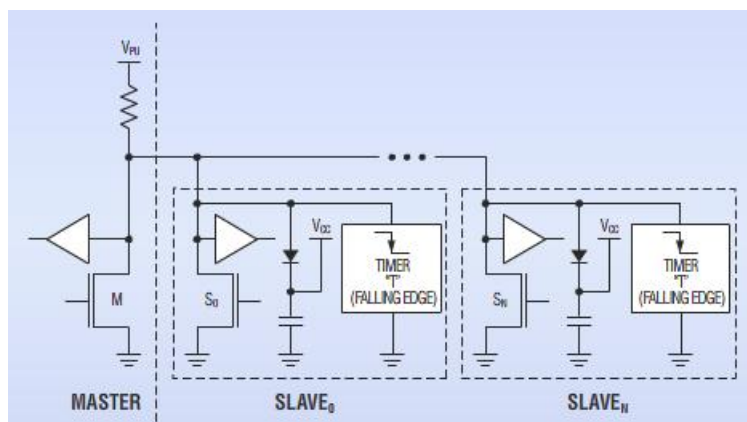


Figura 4.6: Barramento *1-Wire*

O protocolo *1-Wire* utiliza níveis lógicos CMOS/TTL (*Complementary Metal Oxide Semiconductor / Transistor Transistor Logic*) onde o nível lógico zero é representado com o valor máximo de 0,8V e o nível lógico um é representado com o valor mínimo de 2,2V.



Figura 4.7: Níveis lógicos CMOS/TTL

Tanto o *master* como os *slaves* são transceptores (transmissores-receptores), permitindo que a comunicação possa ser feita em ambos os sentidos, de maneira sequencial, porém a comunicação só pode ser feita num sentido de cada vez (*half-duplex*).

É utilizada uma resistência de *pull up* de $1,5K\Omega$ por forma a controlar a corrente na linha de dados, tornando assim possível a transição do nível lógico zero para o

nível lógico um [39]. Para efectuar a transição do nível lógico um para zero é utilizado um MOSFET (*Metal Oxide Semiconductor Field Effect Transistor*) comandado pelo *master*, tal como pode ser visto na figura 4.6.

Para escrever um valor lógico um o *master* leva a linha de dados a zero durante aproximadamente $8\mu s$ e para escrever o valor lógico zero o *master* leva a linha de dados a zero durante aproximadamente $60\mu s$.

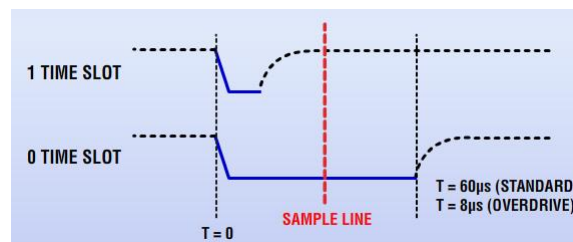


Figura 4.8: Escrever zero ou um em *1-wire*

No início das comunicações o *master* não conhece os seus *slaves* então, torna-se necessário um processo de identificação. Para começar, o *master* leva a linha ao nível lógico zero durante pelo menos $480\mu s$, obrigando os *slaves* que estejam na linha a fazer *reset* [37]. Os *slaves* ligados à linha de dados reconhecem a operação e respondem ao *master* enviando um pulso de presença. Com esta operação o *master* apenas sabe que existe pelo menos um dispositivo na linha. De seguida, o *master* acede ao código de identificação de um determinado escravo, configurando-o, podendo depois comunicar com ele uma vez que já conhece o seu número de série. Na seguinte figura podemos ver estas três fases.

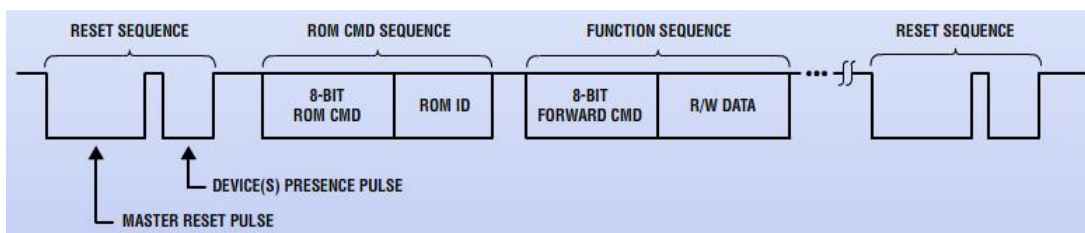


Figura 4.9: Sequência de transmissão *1-Wire*

4.3.2 Alimentação dos Dispositivos

Este protocolo tem o nome de *1-Wire* pois a transferência de dados é feita apenas por um condutor, isto é possível uma vez que, por convenção o condutor de referência não é considerado. Os dispositivos podem ser conectados ao barramento *1-Wire* de duas maneiras: alimentação parasita ou alimentação externa [40].

4.3.2.1 Alimentação Parasita

Na alimentação parasita os *slaves* são alimentados pela linha de dados. Isto é possível pois cada escravo tem um condensador que é carregado quando a linha de dados se encontra no valor lógico um sendo o escravo simultaneamente alimentado pela linha. Quando o valor na linha de dados transita para o valor lógico zero o escravo deixa de ser alimentado pelo barramento e passa a ser alimentado pelo condensador que tinha sido carregado anteriormente. Esta forma de ligar os dispositivos faz com que não seja necessário o condutor de alimentação externa, sendo apenas necessário dois condutores: o de linha de dados e o de referência. Este tipo de alimentação pode ter alguns problemas quando os *slaves* se encontram muito longe do *master*, ou quando existe uma sequência de dados com muitos zeros.

4.3.2.2 Alimentação Externa

Na ligação com alimentação externa, os *slaves* são alimentados não pelo barramento mas sim por uma fonte externa situada próxima destes. A tensão de alimentação igual a 5V. A desvantagem deste tipo de ligação, reside na presença de um terceiro condutor e uma fonte de alimentação externa. Tem, no entanto, a vantagem de se poder aumentar o número de dispositivos na linha.

4.3.3 Topologias das redes 1-Wire

As redes 1-Wire podem ter várias formas, sendo qualificadas pela disposição dos seus *slaves* e pela distância que estes podem ter relativamente ao *master*. Assim a *Dallas Semiconductor* estabeleceu três topologias: linear, ramificada e em estrela [40] [36].

4.3.3.1 Topologia linear

Neste tipo de topologia o *master* está ligado em linha aos *slaves*. Podem ainda existir pequenas ramificações desde que não sejam maiores que três metros.

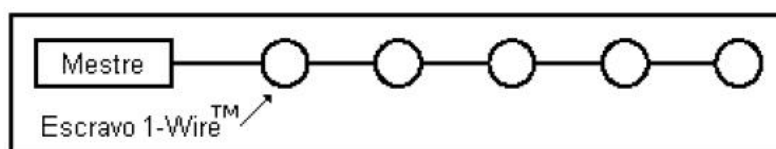


Figura 4.10: Topologia linear

4.3.3.2 Topologia ramificada

Na topologia ramificada, a rede tem uma linha principal que vai até ao último *slave* da rede. Desta linha saem ramificações com um tamanho superior a três metros. Esta topologia tem o inconveniente de as ramificações criarem impedância nos pontos de ligação.

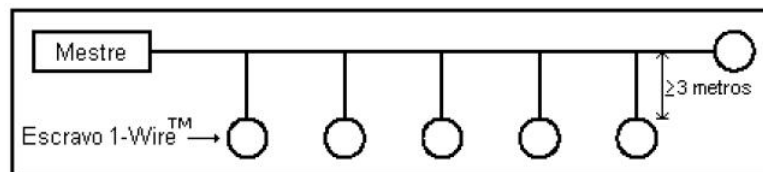


Figura 4.11: Topologia ramificada

4.3.3.3 Topologia em estrela

A topologia em estrela é bastante diferente das duas anteriores pois tem ramificações perto do *master*. Os *slaves* podem ser ligados no fim ou ao longo dessas ramificações. Esta topologia tem o inconveniente das ramificações criarem uma impedância muito elevada perto do *master* podendo haver problemas com reflexões. Existem dispositivos que permitem seleccionar as ramificações fazendo com que cada uma delas seja activada de forma independente. Não é aconselhada a utilização desta topologia sem este tipo de dispositivos, pois torna-se pouco fiável.

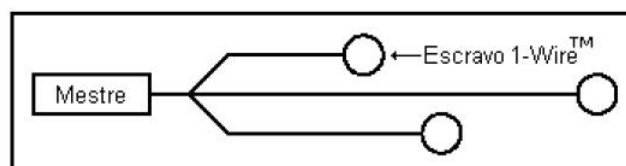


Figura 4.12: Topologia em estrela

4.3.4 Tipos de dispositivos

Hoje em dia existe uma grande variedade de dispositivos *1-Wire* com várias funcionalidades, podemos destacar:

- Memórias EPROM, EEPROM, ROM. Alguns exemplos destes dispositivos são o DS2505 EPROM de 16k, o DS2431 EEPROM de 1K e o DS2411 ROM de 64 *bits*;
- Sensores de temperatura e *switches*. Podemos salientar DS1822 e DS1820 como termómetros digitais e DS2408 que é um *switch* de 8 canais;

- Produtos de interface *1-Wire*. Nestes podemos destacar o DS2482-100 que é um *master* com 1 canal *1-Wire*, o DS2482-800 que é um *master* com 8 canais *1-Wire* e o DS2480B que é um *line drive* (série *1-Wire*);
- Conversores analógico digital como o DS2450 que é uma ADC de 4 canais com resolução programável;
- Relógios de tempo real e contadores de tempo. Por exemplo DS2422 é um *data logger* de temperatura que contém ainda uma ADC e o DS2417 que permite contar o tempo e funciona como calendário;
- Protectores, selectores e monitores de bateria tais como o DS2438 que é um monitor de bateria que permite fazer a leitura da corrente, da tensão e da temperatura e o DS2762 que é um monitor e protector de bateria.

Podem destacar-se ainda outros dispositivos *1-Wire* sendo de referir o DS2409 que é um dispositivo que permite seleccionar ramificações de redes *1-Wire* de forma independente sendo essencial para criar e controlar várias redes deste tipo e o DS9490 que é um adaptador USB *1-Wire*. Os *datasheet* de todos estes dispositivos podem ser encontrados [41].

4.3.5 Avaliação de Desempenho de *1-Wire*

O Barramento *1-Wire* proporciona uma forma de comunicação lenta mas de baixo custo, uma vez que os dispositivos utilizados são baratos e requerem menos cabelagem. Este protocolo tem algumas limitações como uma distância máxima de 150 metros e uma capacidade máxima de carga não aconselhada a mais de 20 dispositivos. Estas limitações podem ser minoradas de três formas: utilizando dispositivos para comutar barramentos (DS2409), escolhendo a alimentação mais adequada e a melhor topologia para a aplicação em questão [38].

Capítulo 5

Controlo Difuso

Nos últimos anos tem vindo a aumentar o interesse na utilização de sistemas inteligentes em algoritmos de controlo. Dentre várias técnicas que utilizam estes sistemas, podemos destacar a lógica difusa. Esta tem uma grande aplicabilidade, especialmente, nas áreas de controlo e tomada de decisão. O ponto forte do controlo difuso deriva da sua capacidade de inferir conclusões e gerar respostas baseadas em informações qualitativas, vagas e ambíguas. Esta teoria pretende traduzir de forma matemática o conhecimento qualitativo adquirido pelos seres humanos com a sua experiência, tornando-se óptima para sistemas de difícil modelação e não lineares. Neste aspecto, os sistemas de controlo difuso são extremamente poderosos tendo a capacidade de raciocinar de forma semelhante à dos humanos. Destacam-se inúmeros exemplos de controladores difusos como transmissões automáticas, ABS, *cruise control*, câmaras fotográficas e sistemas de ventilação. Um controlador difuso é constituído por três blocos: fuzificação, inferência e desfuzificação [3] [42].

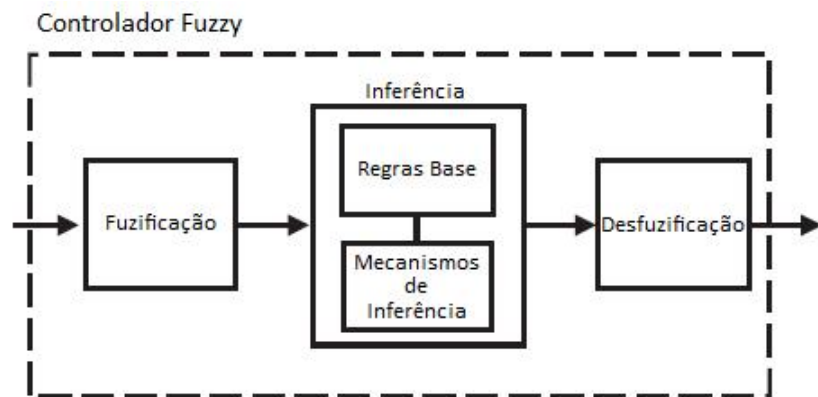


Figura 5.1: Controlador difuso

5.1 Fuzificação

Os valores de entrada do controlador têm de ser normalizados num universo de discurso pré-definido, uma vez que, o controle difuso não utiliza variáveis numéricas mas sim variáveis linguísticas. Assim, quando existe um valor de entrada este é convertido em graus de pertença relativamente aos n conjuntos difusos existentes no seu universo de discurso. Como pode ser visto na figura 5.2 as funções de pertença difusas podem ser do tipo triangular ou trapezoidal [42].

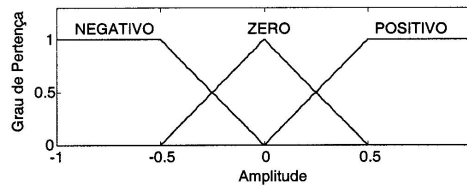


Figura 5.2: Universo de discurso e funções de pertinência

5.2 Inferência

O bloco de Inferência é muito importante pois é ele que contém as regras difusas e é nele que é aplicada a tomada de decisão. Este bloco está então dividido em dois sub-blocos: base de regras e mecanismo de inferência.

5.2.1 Base de Regras

A base de regras é constituída por um conjunto de regras difusas que descrevem o sistema a controlar. As regras difusas são do tipo *If x Then y*, no qual x e y são proposições difusas. As proposições podem ser simples ou ligadas com conectivas lógicas do tipo “and” ou “or”. As regras difusas podem assim ser simples do tipo *If x_1 is A_1 Then y is B* , ou ligadas por conectivas lógicas do tipo *If x_1 is A_1 and x_2 is A_2 Then y is B* . Estas regras são fundamentais para o controle difuso pois se não forem elaboradas correctamente o controlador terá um mau desempenho. Existem várias formas de elaborar regras difusas [43]:

- Utilizando a experiência do operador e do engenheiro de controle, podendo isto ser feito de forma heurística ou com realização de questionários;
- Utilizando uma descrição linguística da dinâmica do sistema a controlar;
- Utilizando auto-aprendizagem, no qual o sistema tem a capacidade de criar e modificar as regras baseando-se na sua experiência.

5.2.2 Mecanismo de inferência

O mecanismo de inferência é a parte mais importante do controlador difuso pois permite que o controlador tenha uma capacidade de raciocínio semelhante à dos humanos. Este mecanismo possibilita inferir a acção de controlo a aplicar partindo de uma base de regras. Quando a inferência recebe um valor determina para cada regra os graus de pertença correspondentes. Utilizando estes graus de pertença e uma função de agregação determina-se um novo conjunto difuso que traduz o grau de pertença da variável de controlo. Existem vários mecanismos de inferência sendo os mais comuns o *max-min* e o *max-prod* [43] [44].

Um exemplo de inferência pode ser visto na figura 5.3.

Considerando que A_1 , A_2 , B_1 , B_2 , C_1 , C_2 são conjuntos *difusos*, x_1 e x_2 são parâmetros de entrada e y parâmetro de saída, podemos definir duas regras difusas R_1 e R_2 da seguinte forma:

$R_1 : \text{If } x_1 \text{ is } A_1 \text{ and } x_2 \text{ is } B_1 \text{ Then } y \text{ is } C_1$

$R_2 : \text{If } x_1 \text{ is } A_2 \text{ and } x_2 \text{ is } B_2 \text{ Then } y \text{ is } C_2$

Se for considerado um valor de entrada representado por x_1 , pode ser visualizado o seu grau de pertença relativo a todos os grupos pela intercepção da recta vertical com os conjuntos difusos. Neste caso em concreto, x_1 tem grau de pertença diferente de zero nos dois conjuntos, A_1 e A_2 . O mesmo pode ser observado com o valor representado por x_2 verificando-se que este tem grau de pertença diferente de zero nos dois conjuntos, B_1 e B_2 . Em R_1 e R_2 é utilizada a conectiva lógica “and” e como no controlo difuso esta tem o significado de mínimo dos conjuntos, assim é seleccionado o menor grau de pertença. Isto pode ser observado na figura 5.3, pelas linhas horizontais em cada uma das regras. Em seguida é aplicado o operador “max” para a agregação das regras. Outro exemplo de inferência encontra-se representado na figura 5.4, em que o pensamento lógico é o mesmo mas em vez de se utilizar o operador “min” é utilizado o “prod” que representa a implicação segundo Larsen.

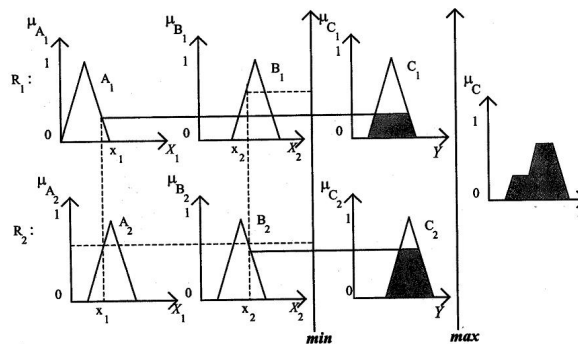


Figura 5.3: Representação gráfica do mecanismo de inferência *max-min*

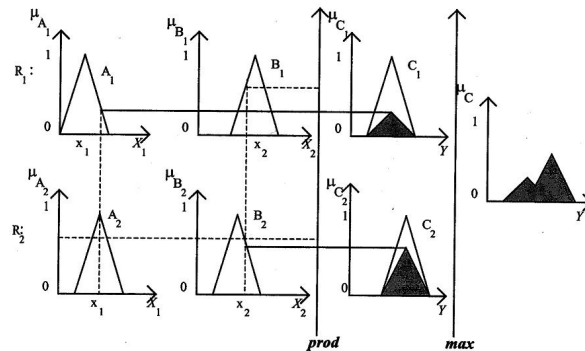


Figura 5.4: Representação gráfica do mecanismo de inferência *max-prod*

5.3 Desfuzificação

A desfuzificação é o processo de conversão dos valores difusos em valores numéricos. O conjunto difuso vindo do bloco de inferência entra no bloco de desfuzificação onde lhe é aplicada uma estratégia de desfuzificação convertendo-o num valor numérico. Este valor representa a distribuição de possibilidades de uma determinada acção de controlo. Existem várias estratégias de desfuzificação sendo as mais conhecidas o critério do máximo, a média do máximo, a altura e o centro de área, sendo este último o mais utilizado [3].

5.4 Controlo Difuso em Sistemas Embutidos

Embora os conceitos base da lógica difusa não sejam difíceis de compreender, não é fácil pô-la em prática pois requer bastante poder de cálculo. Deste modo, para sistemas com pouco poder de cálculo como microprocessadores e microcontroladores (sistemas embutidos em geral) torna-se necessário a utilização de algoritmos que ultrapassem esta limitação, assim, uma solução possível é a utilização de um controlador difuso baseado numa tabela. Este requer muito menos poder computacional pois a forma das funções de pertinência, o método de inferência e o método de desfuzificação deixam de ser feitos em tempo real, para ser feitos *offline* no qual é criada uma tabela com a relação entre a entrada do sistema e a saída do controlador difuso. Como a tabela não pode conter todos os valores de relação entrada/saída, uma vez que necessitaria uma infinidade de pontos, existem erros de quantificação. Este tipo de erros pode ser minimizado utilizando interpolação entre pontos ou aumentando o número de valores na tabela, assim, este tipo de erros não se torna muito relevante no algoritmo de controlo [42].

Capítulo 6

Implementação

Neste projecto foi desenvolvido um sistema de aquisição, monitorização e controlo para aplicações meteorológicas e de engenharia agrícola. Mais concretamente foi criado um sistema de controlo distribuído com a utilização conjunta de um PC , uma plataforma TINI, sensores *1-Wire* e linguagem Java. A seguinte figura mostra uma pequena estação de controlo de irrigação com acesso local e remoto sob a forma de um diagrama de blocos.

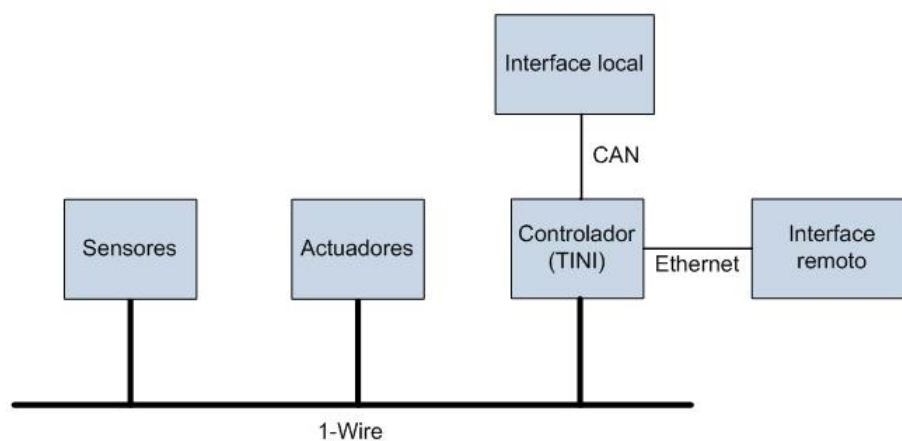


Figura 6.1: Sistema de controlo de irrigação

6.1 Sensores

No desenvolvimento deste projecto foram utilizados vários sensores: temperatura, humidade do ar, humidade do solo, radiação solar e medidor de chuva.

Tal como foi referido anteriormente toda a informação sobre os dispositivos *1-Wire* pode ser obtida em [41].

6.1.1 Sensor de Temperatura

Para medir a temperatura foi utilizado o dispositivo *1-Wire* DS1820. Este é um termómetro digital que indica a temperatura do dispositivo com precisão de *9bits* estando ligado directamente ao barramento *1-Wire*.

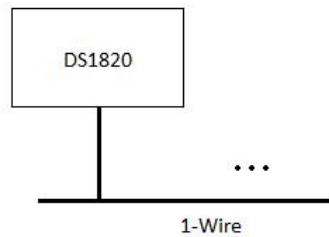


Figura 6.2: Sensor de Temperatura

6.1.2 Sensor de Humidade do ar

Para medir a humidade do ar foi conjugado o dispositivo *1-Wire* DS2438 com o sensor de humidade HIH-3610. Mais informação sobre o sensor de humidade HIH-3610 podem ser consultadas em [45]. Este sensor apresenta na saída uma tensão linear com a humidade. O DS2438 é um monitor de baterias que contém uma ADC de *10bits* e pode ser utilizado como sensor de temperatura, medidor de corrente e medidor de tensão. A saída do sensor HIH-3610 está ligada à ADC do DS2438 que, por sua vez, se encontra ligado ao barramento *1-Wire* como pode ser visto na figura 6.3. Nesta figura também pode ser observado o esquema eléctrico deste sensor e uma fotografia deste sensor.

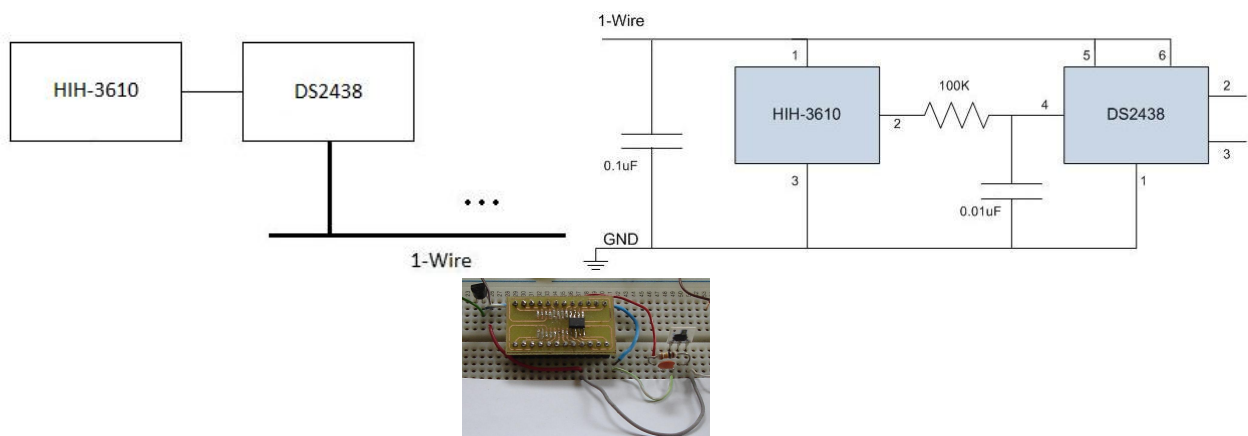


Figura 6.3: Sensor de humidade do ar implementado

6.1.3 Sensor de Humidade do solo

Para medir a humidade do solo foi utilizado um sensor Watermark em conjunto com o dispositivo *1-Wire* DS2450. O DS2450 é uma ADC de 4 canais com escala máxima de 5V e com uma precisão programável. Como foi referido no capítulo 3, o Watermark é um sensor de humidade resistivo que tem de ser alimentado com uma tensão AC. Tornou-se assim necessário a implementação de um circuito de alimentação e um circuito para efectuar leituras do sensor por parte do DS2450. Tal como pode ser visto na figura 6.4, para alimentar o divisor resistivo onde se encontra o sensor Watermark foi utilizado um circuito oscilador com uma frequência de 1kHz, existindo assim uma diferença de tensão entre a alimentação e o ponto intermédio do divisor resistivo permitindo assim saber o valor da resistência do sensor. Por forma às entradas da ADC do DS2450 poderem medir correctamente os valores da alimentação e o valor intermédio, torna-se necessário converter a tensão AC em tensão contínua. Para fazer esta conversão foram utilizados circuitos detectores de pico e para isolar os circuitos foram utilizados dois *Buffers*.

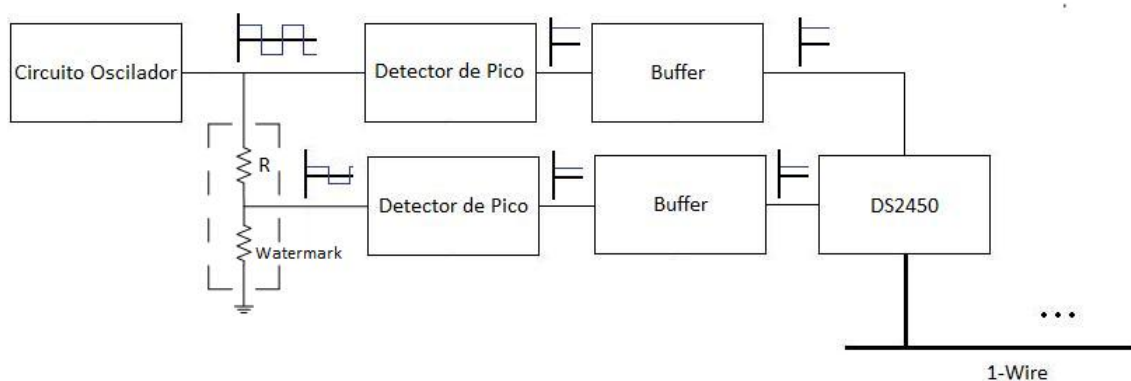


Figura 6.4: Sensor de humidade do solo implementado

O esquema eléctrico pode ser visto na figura 6.5 e na figura 6.6 pode ser observado a fotografia do sensor e a electrónica associada.

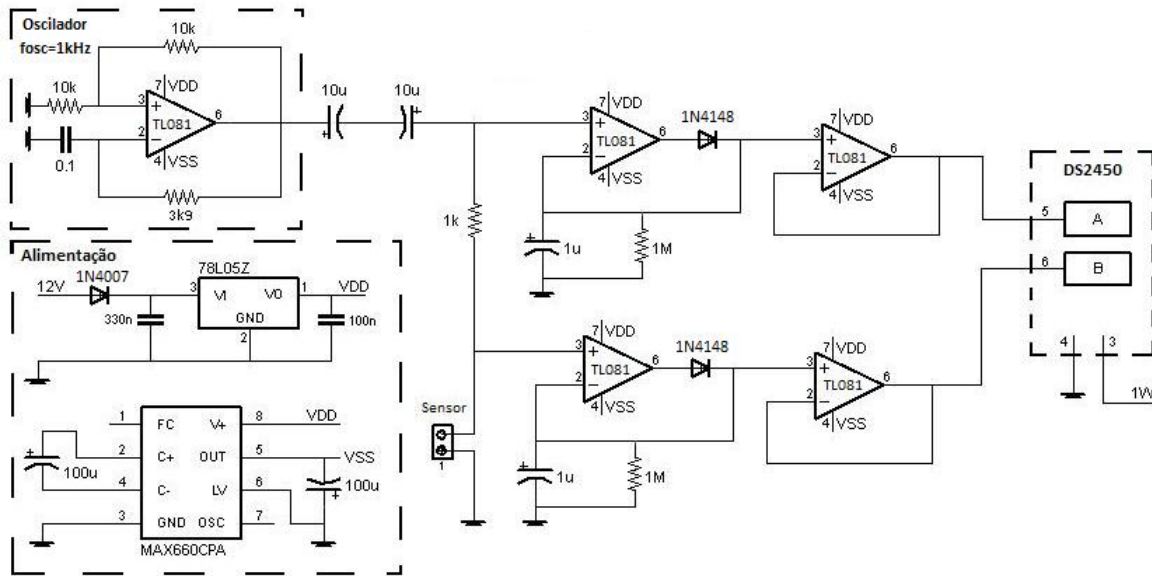


Figura 6.5: Esquema eléctrico do sensor de humidade do solo

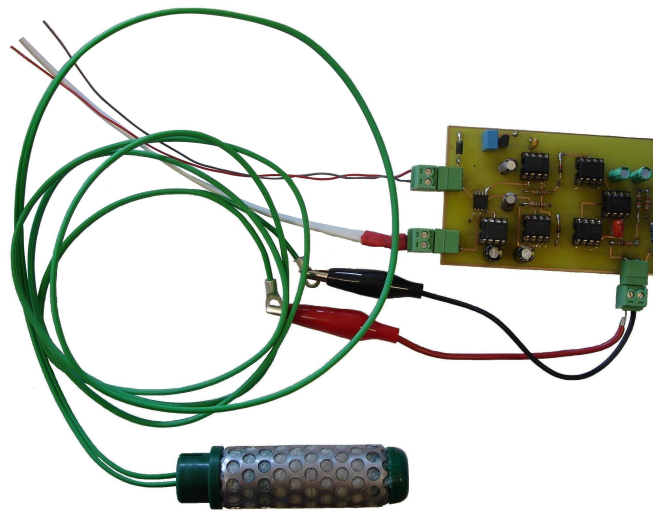


Figura 6.6: Fotografia do circuito e sensor de humidade

Como o valor medido é uma resistência, torna-se necessária a conversão desta para valores de humidade. Para o efeito foi utilizada a curva de calibração da figura 3.1 que foi traduzida nas equações 6.1 e implementadas no software desenvolvido para a plataforma TINI. Tal como a curva de calibração estas equações foram retiradas do trabalho [46]. Nestas equações a variável ohms representa a resistência do sensor e a variável kPa representa o valor de humidade calculado em kiloPascal.

$$\left\{ \begin{array}{ll} kPa = \frac{(ohms-550)}{50} & 550 \leq ohms \leq 1000 \\ kPa = 9 + \frac{(ohms-1000)}{100} & 1000 \leq ohms \leq 1100 \\ kPa = 10 + \frac{(ohms-1100)}{180} & 1100 \leq ohms \leq 2000 \\ kPa = 15 + \frac{(ohms-2000)}{200} & 2000 \leq ohms \leq 6000 \\ kPa = 35 + \frac{(ohms-6000)}{160} & 6000 \leq ohms \leq 9200 \\ kPa = 55 + \frac{(ohms-9200)}{150} & 9200 \leq ohms \leq 12200 \\ kPa = 75 + \frac{(ohms-12200)}{135} & 12200 \leq ohms \leq 15575 \end{array} \right. \quad (6.1)$$

6.1.4 Sensor de Radiação solar

Para o sensor de radiação foi conjugado um painel solar e um DS2438 configurado como medidor de tensão. Como a saída do painel solar é em corrente foi utilizado um divisor resistivo por forma a obter uma relação entre a corrente e a tensão permitindo efectuar uma leitura directa por parte da ADC do DS2438. Por forma a isolar os circuitos foi utilizado um *buffer*. O sensor implementado pode ser observado na figura 6.7, o esquema eléctrico na figura 6.8 e a fotografia na figura 6.9.

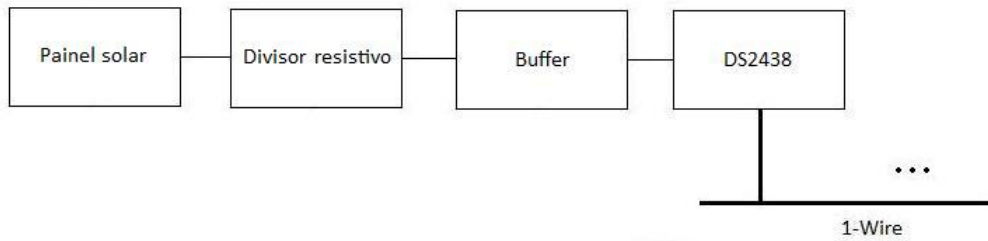


Figura 6.7: Sensor de radiação solar (Diagrama de blocos)

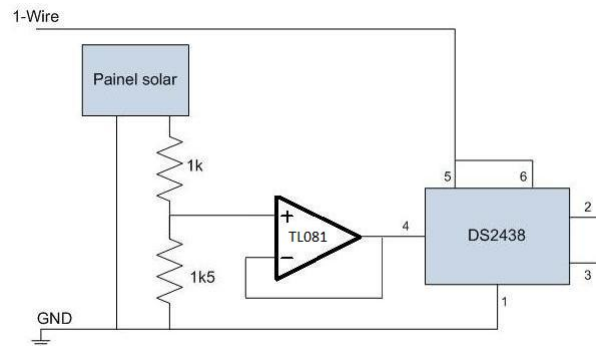


Figura 6.8: Sensor de radiação solar (Esquema eléctrico)

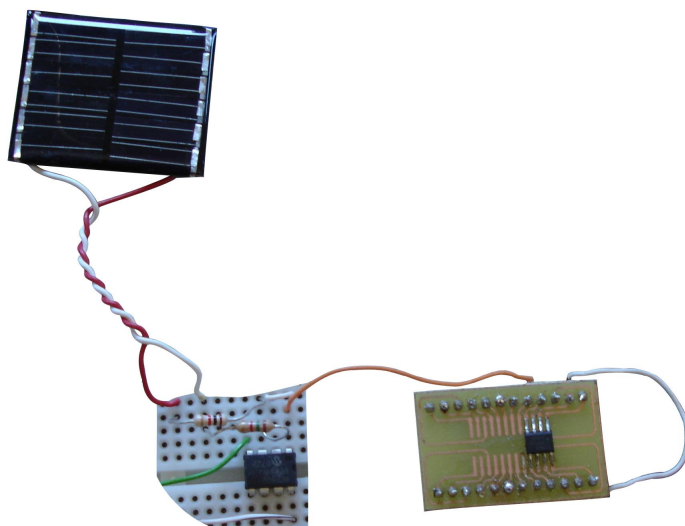


Figura 6.9: Sensor de radiação solar (Fotografia)

6.1.5 Medidor de Chuva

Para o medidor de chuva foi utilizado um DS2423 e um recipiente afunilado que contém um balancê com um *reed switch*. O DS2423 é um contador *1-Wire* que permite medir os eventos gerados pelo *reed switch*. Sabendo que cada lado do balancê suporta 0.33mm de água basta multiplicar este valor pela contagem efectuada pelo DS2423 para saber a quantidade de precipitação. A figura 6.10 apresenta um diagrama de blocos do medidor de chuva implementado e a figura 6.11 apresenta a fotografia do medidor de chuva.

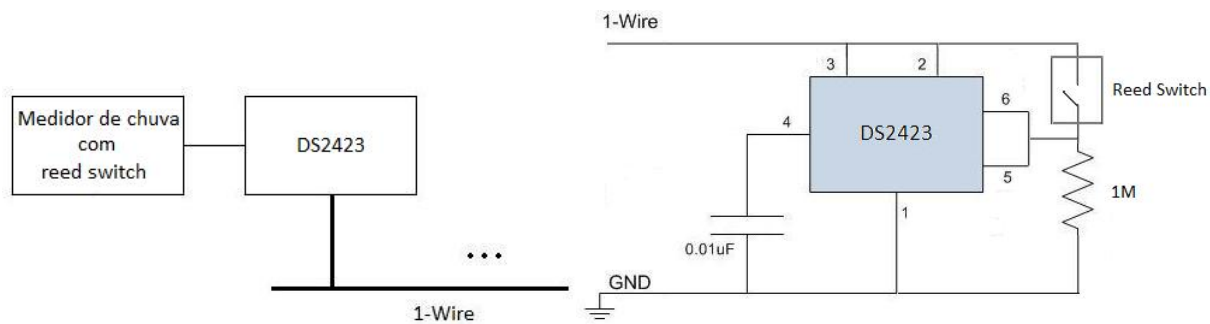


Figura 6.10: Medidor de chuva



Figura 6.11: Balde afunilado e o balancé com um *reed switch*

6.2 Actuador

Tal como foi referido anteriormente os actuadores num sistema de irrigação são electroválvulas, no entanto, devido à falta de água corrente no laboratório e tentando não a desperdiçar, foi utilizado uma pequena bomba de água dentro de um recipiente com água para servir de actuador.

No âmbito do sistema desenvolvido, a actuação tem como objectivo aumentar a percentagem de humidade no solo.

Deste modo, para a implementação do actuador, foi efectuada uma utilização conjunta de uma bomba de água com um módulo de potência; a bomba de água com a função de regar e o módulo de potência para permitir controlar a aplicação da tensão da rede à mesma. O esquema deste módulo é apresentado na Figura 6.12.

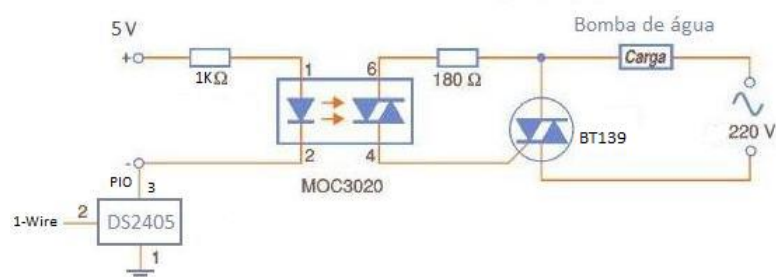


Figura 6.12: Circuito implementado para o actuador

Este módulo contém um *switch* (DS2405) que permite ser controlado remotamente através do barramento *1-Wire*. Por sua vez quando activado, o *switch*, fecha o circuito

do *led* do foto-triac com isolamento óptico (MOC3020) que, por sua vez, faz conduzir o opto-triac nele contido.

Uma vez que o foto-triac suporta apenas correntes baixas é necessário usar um triac, activado pelo primeiro, para superar esta limitação; o triac, que funciona como um interruptor, entra em condução quando é aplicado um sinal na sua *gate*. A mesma cessa quando esse sinal deixa de ser aplicado e ocorre uma passagem do sinal da rede por zero.

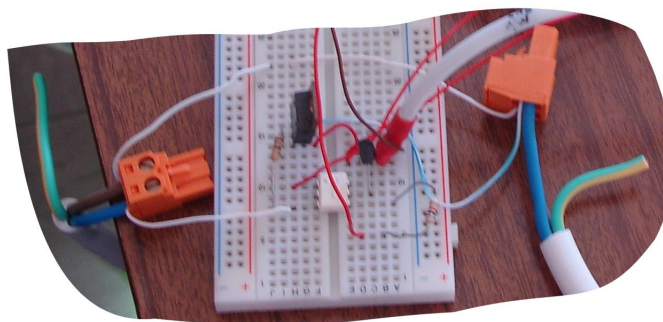


Figura 6.13: Módulo de potência do actuador

6.3 Interface Local

Para a realização do interface local foi utilizado um microcontrolador PIC182580, um conjunto de 24 botões em matriz formando um teclado, um LCD de duas linhas e quarenta colunas e um *buzzer*. Este módulo foi interligado à plataforma TINI utilizando o protocolo CAN. A figura 6.14 apresenta um diagrama de blocos do interface local. O esquema eléctrico pode ser visualizado no apêndice C.

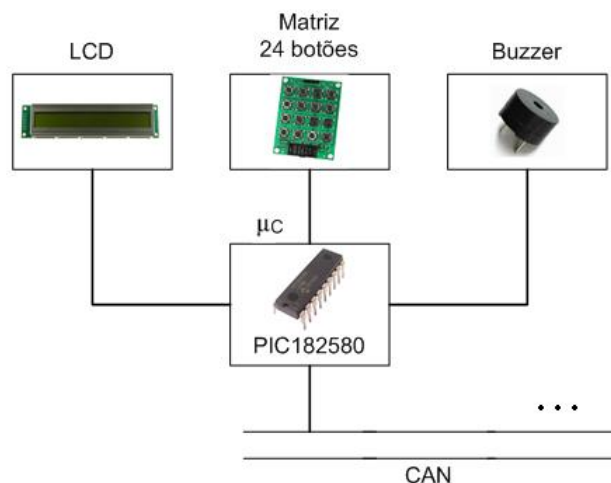


Figura 6.14: Interface local

Para este interface foi desenvolvido um *software* que permite ao utilizador usar o teclado para inserir a hora de início e quanto tempo deve regar, e o LCD serve como visualizador permitindo que o utilizador verifique o correcto funcionamento do controlador. Na figura 6.15 são apresentados os fluxogramas do *software* presente no microcontrolador PIC182580 e na plataforma TINI.

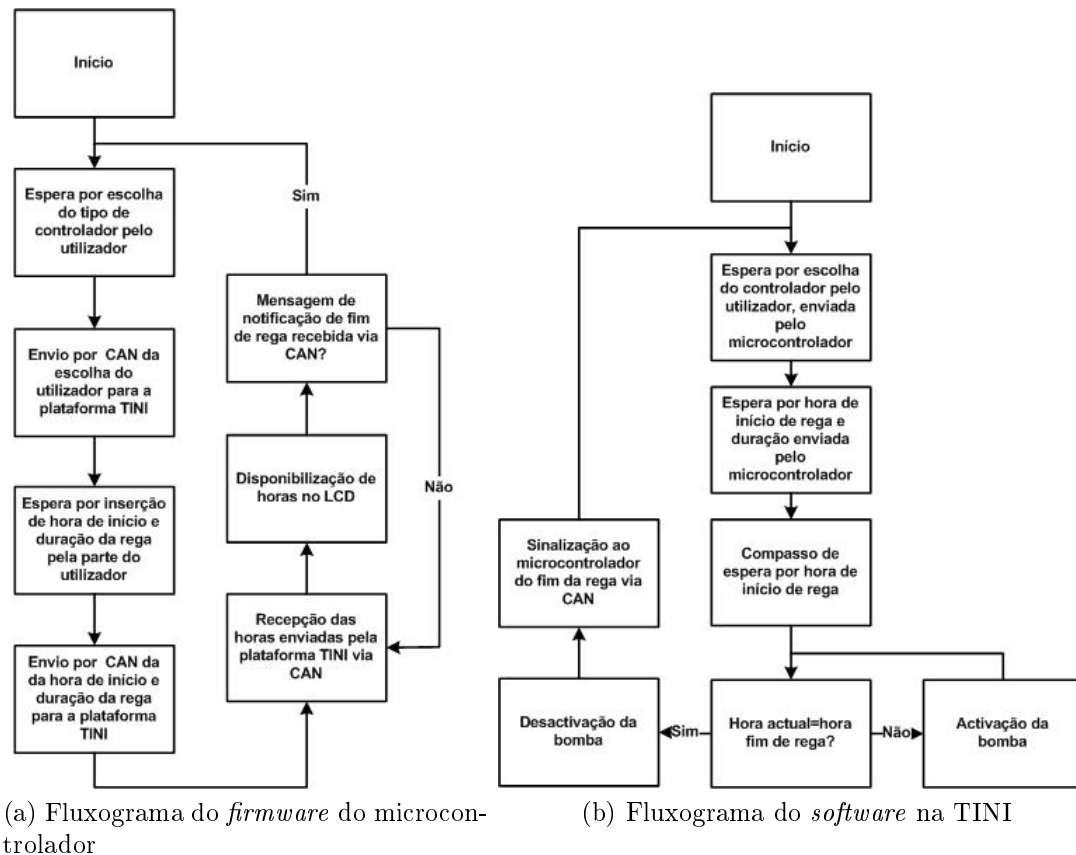


Figura 6.15: Diagrama do código do interface local

O *firmware* presente no microcontrolador começa por pedir ao utilizador que introduza a sua escolha de uma listagem de controladores apresentados, ficando bloqueado à espera da resposta. Quando o utilizador introduz a sua escolha o microcontrolador comunica com a plataforma TINI via CAN informando-a do mesmo. Seguidamente é requerido ao utilizador que introduza a hora de início de rega e a duração da mesma, ficando o microcontrolador mais uma vez bloqueado à espera de resposta. Após resposta do utilizador o microcontrolador envia estes parâmetros via CAN. A partir deste momento o microcontrolador entra num ciclo do qual só sairá quando da recepção de uma mensagem CAN sinalizando o fim de rega. Durante a execução deste ciclo a hora actual será continuamente recebida via CAN e disponibilizada no LCD. No fim do ciclo retorna ao estado inicial, onde requer novamente ao utilizador que introduza a sua

escolha de controlador.

O *software* presente na plataforma TINI inicia a sua actividade esperando a notificação, pela parte do microcontrolador, da escolha do controlador e a hora de início de rega e duração da mesma. De seguida cria um compasso de espera até que a hora actual coincida com a hora de início de rega. A partir deste momento activa a bomba até que o período de rega termine. No fim da rega informa o microcontrolador da sua ocorrência retornando ao estado inicial de espera pelos parâmetros.



Figura 6.16: Fotografias do interface local e circuito eléctrico

6.4 Interface Remoto

Para a elaboração do interface remoto foi utilizado um computador com MATLAB e a plataforma TINI. A ligação foi feita utilizando *sockets* permitindo uma comunicação via *Ethernet* entre a plataforma TINI e o MATLAB. Neste último foi criada uma aplicação gráfica que permite seleccionar vários controladores diferentes bem como a visualização gráfica dos sensores monitorizados. Na figura 6.17 pode ser observado um diagrama de blocos do interface remoto.

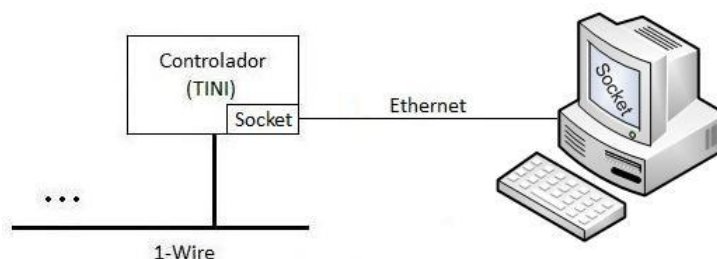


Figura 6.17: Interface remoto

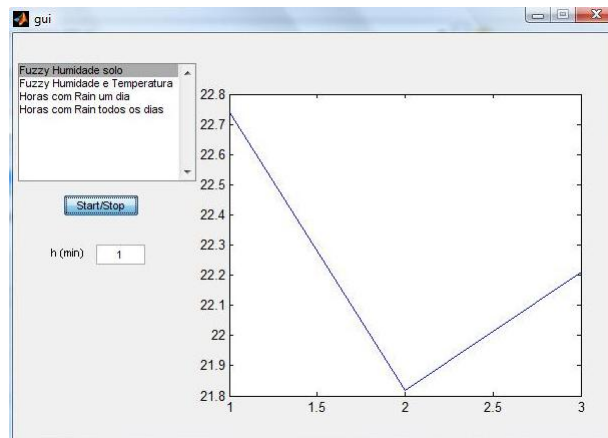


Figura 6.18: Plataforma gráfica no MATLAB

Capítulo 7

Resultados e conclusões

Neste capítulo são apresentados os resultados obtidos na implementação de uma malha que visa o controlo de um processo de irrigação, baseada na utilização conjunta de sensores *1-Wire*, linguagem Java e algoritmos de controlo difuso.

Para este efeito, existem várias filosofias possíveis de implementação que se podem basear nos mais variados tipos de sensores, tais como, por exemplo, sensores de humidade do ar, humidade do solo, temperatura, luminosidade, entre outros. A malha de controlo implementada no âmbito desta dissertação, utiliza um sensor de humidade do solo. A escolha deste tipo de sensor em detrimento dos restantes deve-se ao facto de a variável mensurável pelo mesmo ser umas das que mais influência apresenta neste tipo de sistemas.

O sensor de humidade do solo utilizado é do tipo resistivo, o que implica que o valor da humidade do solo é obtido por medida indirecta. Ou seja, com este tipo de sensor é possível medir-se uma resistência variável com a humidade do solo e a partir desta medida é possível obter-se o valor da mesma utilizando, para o efeito, uma curva de calibração. A curva de calibração representa, portanto, uma equação matemática que fornece uma relação entre o valor de resistência, medido pelo sensor e a humidade do solo.

O valor de resistência medido neste sensor pode ser influenciado, igualmente, pela temperatura. Existem algumas formas de efectuar uma compensação deste efeito. Para a utilização deste sensor na malha de controlo surgiu a necessidade de efectuar uma caracterização do mesmo em termos de dinâmica e de sensibilidade ao factor temperatura. Esta caracterização serve o propósito de verificar qual deverá ser o tempo de amostragem do algoritmo de controlo a utilizar e a necessidade de compensação da temperatura.

7.1 Caracterização do sensor de humidade do solo

A dinâmica do sensor de humidade do solo está associada à rapidez com que a resistência, medida no mesmo, varia face à transição do estado seco para húmido e vice-versa. De notar que estas duas transições apresentam, tipicamente, tempos bastante diferentes pelo que deve ser feita uma caracterização individual. Para obtenção do valor de humidade a partir da resistência medida é utilizada a curva de calibração descrita pelas equações 6.1 presente no capítulo 6.

7.1.1 Transição do estado seco para o estado húmido

Para verificação da dinâmica nesta transição colocou-se o sensor seco num recipiente cheio de água. O valor do sensor foi monitorizado com uma periodicidade de 6 segundos. Os resultados desta monitorização são apresentados na figura 7.2. Verificou-se que o tempo de reacção na passagem do estado seco para húmido é de aproximadamente 30 segundos (instante a partir do qual o valor de humidade se aproxima 0kPa).

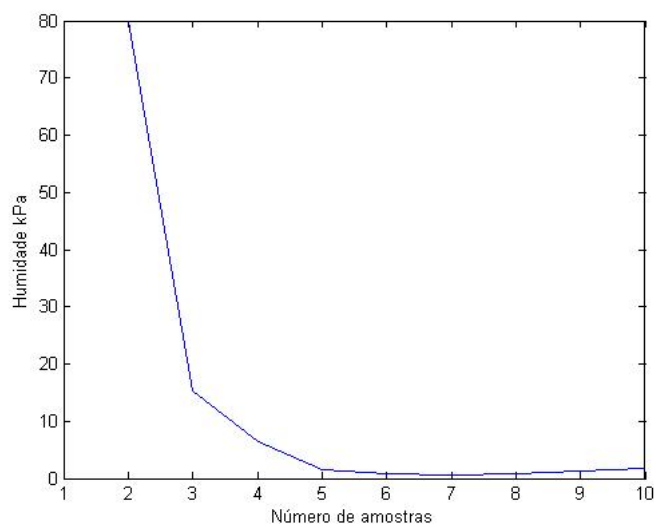


Figura 7.1: Teste Watermark sensor seco para húmido

7.1.2 Transição de estado húmido para seco

Na verificação da dinâmica da transição de estado húmido para seco colocou-se o sensor húmido num recipiente vazio sendo o seu valor monitorizado com uma periodicidade de 5 minutos. Os resultados desta monitorização são apresentados na figura 7.2. Verificou-se que o tempo de reacção na transição do estado húmido para seco é de aproximadamente 30 horas (instante a partir do qual o sensor fica seco).

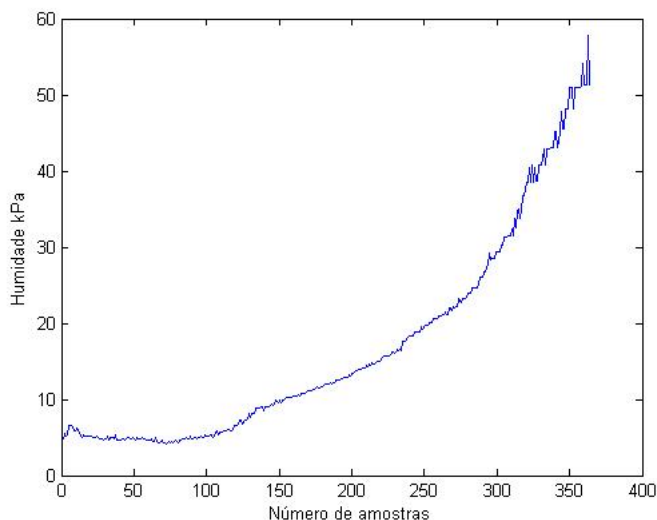


Figura 7.2: Teste Watermark sensor húmido para seco

7.1.3 Sensibilidade ao factor temperatura

Para verificação da sensibilidade do sensor à temperatura, monitorizou-se a mesma durante o processo de secagem com uma periodicidade de 5 minutos. Não foi escolhida a transição oposta dado que, tal como foi anteriormente concluído, esta apresenta um tempo de reacção de 30 segundos, período demasiado curto para obter informação suficientemente rica de variação de temperatura.

Para compensar o efeito provocado pela variação da temperatura pode ser utilizada a equação 7.1 que permite ajustar o valor de resistência medido, esta equação foi retirada do trabalho [46]. Nesta equação a variável T_{0C} corresponde à temperatura em graus celsius e a variável R_{medido} corresponde ao valor de resistência medido em ohms.

$$R_{compensado} = R_{medido} \times (1 + 0.01 \times (1.8 \times T_{0C} + 32) - 75) \quad (7.1)$$

Na figura 7.3 são apresentados os gráficos relativos à resistência medida e à resistência após compensação. Da análise dos mesmos conclui-se que estes são muito idênticos, o que indica que o factor temperatura tem uma influência que poderá ser desprezável. A variação da temperatura pode ser observada na figura 7.4.

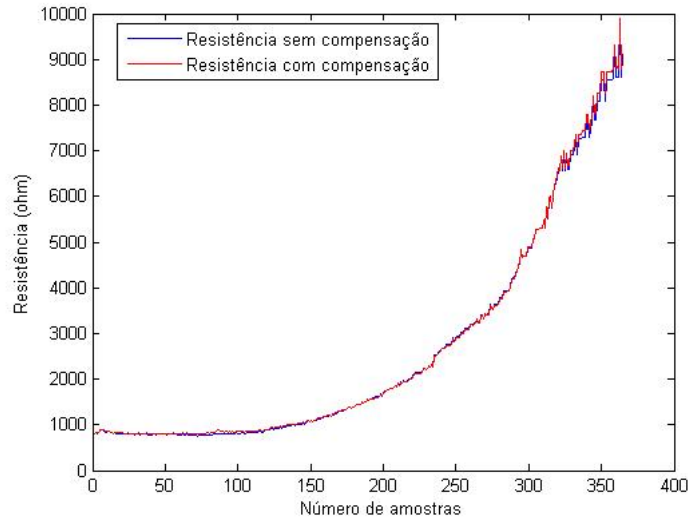


Figura 7.3: Comparação da resistência medida sem compensação e com compensação

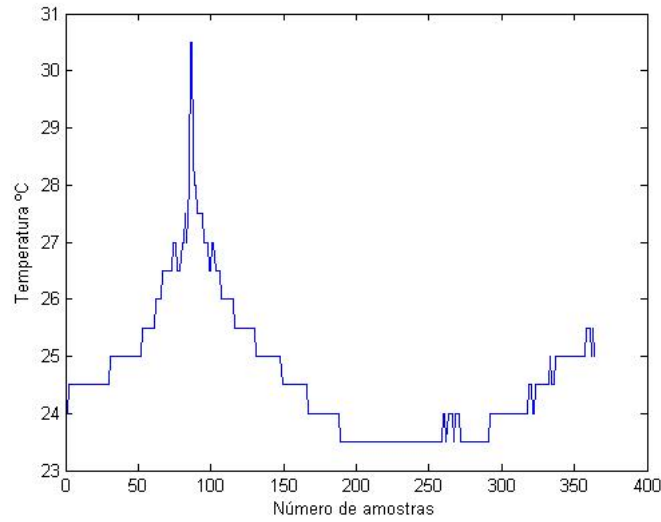


Figura 7.4: Temperatura medida ao longo do teste

Para além da forma de compensação apresentada é, igualmente, possível efectuar uma compensação de temperatura directamente no cálculo do valor de humidade, utilizando para o efeito a equação 7.2. Esta equação tal como a anterior foi retirada do trabalho [46]. Nesta equação a variável T_{0C} corresponde ao valor de temperatura em graus celsius, a variável $K\Omega$ corresponde ao valor de resistência medido em KiloOhms e a variável kPa corresponde ao valor de humidade calculado em KiloPascal.

$$kPa = \frac{(3.213 \times K\Omega + 4.093)}{(1 - 0.009733 \times K\Omega - 0.01205 \times T_{0C})} \quad (7.2)$$

Para uma análise mais detalhada do efeito da temperatura nas medições obtidas pelo sensor de humidade do solo efectuou-se uma comparação entre o valores de humidade obtidos pela curva de calibração com a resistência com e sem compensação e o valor de humidade obtido por compensação directa. Estes resultados são apresentados na figura 7.5.

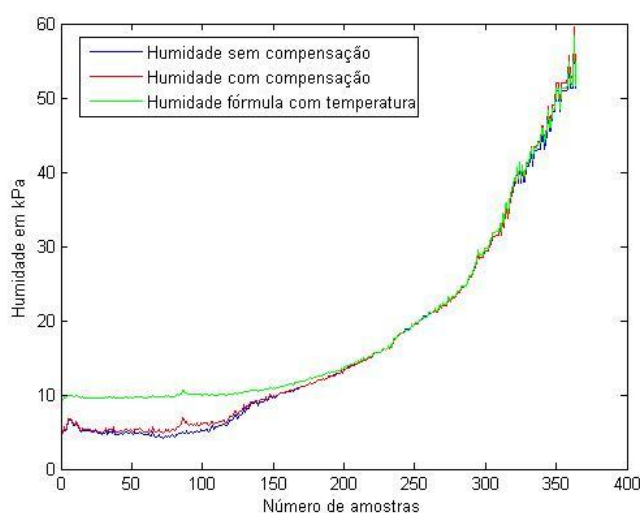


Figura 7.5: Comparação da humidade não compensada com as duas compensações

Desta análise conclui-se que o valor obtido pelos três métodos é muito semelhante. Nota-se uma diferença inicial maior no gráfico relativo à compensação directa (cor verde). Isto deve-se ao facto de a equação utilizada para obtenção desta conversão ter sido concebida para valores de resistência entre os $1\text{k}\Omega$ e $10\text{k}\Omega$ e em casos de humidade muito elevada a resistência medida é bastante inferior a estes valores (tipicamente na ordem dos 700Ω). Assim, conclui-se que o factor temperatura não tem, efectivamente, um efeito muito relevante nos valores de humidade obtidos pelo que não se torna essencial proceder à sua compensação.

7.2 Sistema de controlo

Nesta secção é apresentada a malha de controlo implementada no âmbito desta dissertação. É apresentada a sua estrutura, os parâmetros de controlo utilizados, bem como alguns resultados obtidos.

7.2.1 Estrutura do sistema

Este sistema visa controlar o processo de irrigação de um solo. Para este efeito foi montado segundo a estrutura presente no diagrama de blocos apresentado na figura 7.6.

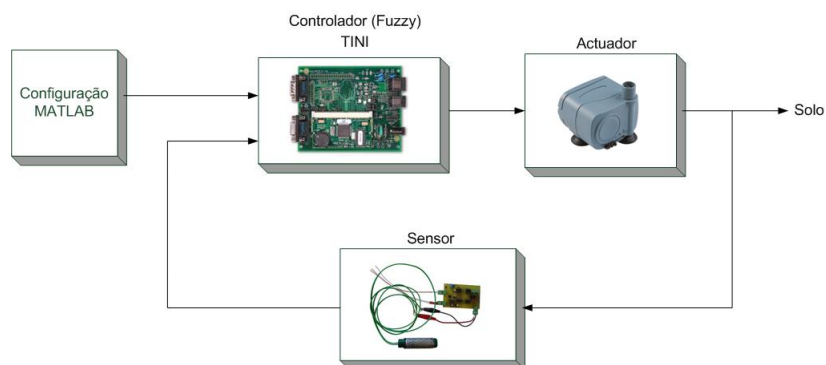


Figura 7.6: Plataforma de Teste

O controlador consiste de uma placa TINI que corre um algoritmo difuso. O tempo de amostragem deste controlador é dado por configurações prévias efectuadas no MATLAB. A aquisição é feita pelo sensor de humidade *Watermark*, anteriormente apresentado, e a actuação por uma bomba de água. O sensor de humidade encontra-se inserido em terra dentro de um vaso. A plataforma de testes é apresentada na figura 7.7.



Figura 7.7: Fotografia da Plataforma de teste

7.2.2 Algoritmo de controlo

O algoritmo de controlo utilizado é um algoritmo difuso. A entrada é a humidade do solo medida pelo sensor e a saída é o tempo durante o qual a bomba de água está ligada durante um periodo de amostragem. A sua parametrização foi efectuada recorrendo ao auxílio da *Fuzzy Logic Toolbox* do MATLAB que permite obter uma relação entre as entradas e saídas do sistema. Mais informações sobre esta *Toolbox* podem ser encontradas em [47]. Os valores utilizados na parametrização foram obtidos de informação fornecida conjuntamente com o sensor *Watermark*. Os conjuntos obtidos para o valor de entrada são apresentados na figura 7.8 conjuntamente com a informação disponibilizada com o sensor *Watermark* [48].

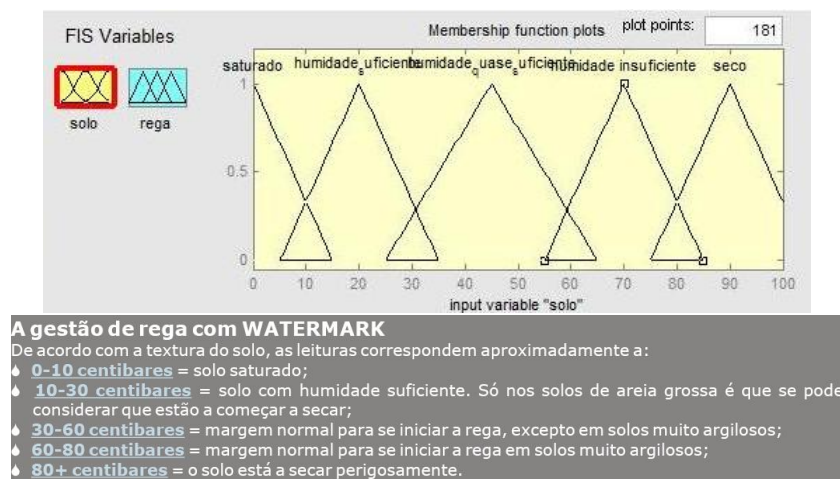


Figura 7.8: Conjuntos difusos do solo

Os conjuntos de saída foram baseados nos valores obtidos na caracterização da dinâmica do sensor. De notar que tem de ser tido em conta o valor correspondente ao tempo de reacção mais baixo, por forma a que o controlador tenha uma dinâmica suficientemente rápida. Destes valores o tempo de reacção mais curto é de 30 segundos, contudo nestes testes o sensor não se encontrava inserido no solo, no caso em que esteja este deverá na realidade ser mais lento. Por este motivo considerou-se um tempo de amostragem de 1 minuto, correspondente ao dobro do valor anteriormente referido. Assim, no algoritmo difuso o tempo de actuação pode variar entre 0 e 1 minuto, tal como apresentado na figura 7.9. Desta forma em cada ciclo de amostragem será atribuído um *duty-cycle* de actuação que poderá variar entre zero e o período de amostragem.

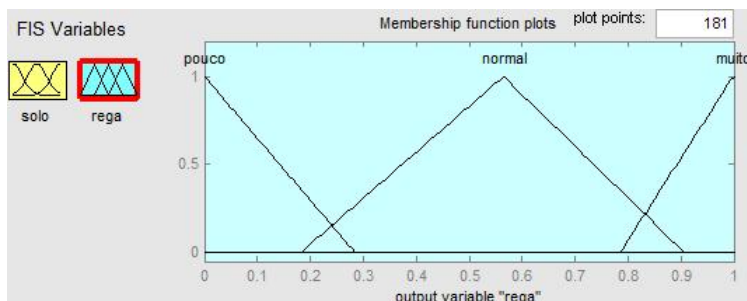


Figura 7.9: Conjuntos difusos de período de rega

De notar que os conjuntos foram escolhidos com base em experiência empírica, podendo não estar otimizados. Para que estes fossem aperfeiçoados seria necessário um conhecimento mais aprofundado do solo, plantas, sistema de irrigação utilizado e caudal em causa.

Com base nestes conjuntos e nas regras apresentadas na figura 7.10 obteve-se no MATLAB uma relação entrada-saída apresentada na figura 7.11.

```

1. If (solo is humidade_suficiente) then (rega is pouco) (1)
2. If (solo is humidade_quase_suficiente) then (rega is normal) (1)
3. If (solo is humidade_insuficiente) then (rega is normal) (1)
4. If (solo is seco) then (rega is muito) (1)
5. If (solo is saturado) then (rega is pouco) (1)

```

Figura 7.10: Regras difusas

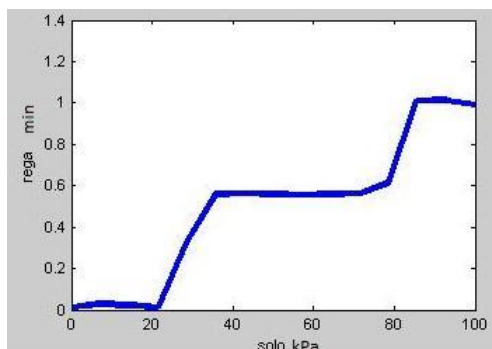


Figura 7.11: Relação entrada saída algoritmo difuso

Dado que este algoritmo de controlo deverá ser executado num sistema embutido, que tem tipicamente recursos reduzidos, criou-se a partir da relação anteriormente obtida a tabela difusa 7.1 que deverá estar presente em memória por forma a que haja uma redução substancial da capacidade de computação exigida pelo algoritmo de controlo.

Humidade do solo (kPa)	Tempo de Rega (min)
0	0
25	0
26	0.13
30	0.39
35	0.56
75	0.56
81	0.65
85	1.0
100	1.0

Tabela 7.1: Tabela difusa

A descrição da relação entrada-saída presente na tabela não é, no entanto, contínua. Pelo que se torna necessária a utilização de uma interpolação linear quando o valor em causa se encontra em pontos intermédios. A variável de saída não pode apresentar valores abaixo de 0.1 dado que isto implica ligar e desligar a bomba com uma rapidez que os barramentos do tipo *1-Wire* não conseguem suportar. Dado isto para valores abaixo de 0.1 o valor da variável de saída é nulo. Tal como pode ser verificado pela análise da tabela 7.1 apenas ocorrerá actuação para valores de humidade superiores a 26kPa, pelo que o valor de humidade do solo deverá tender para o mesmo.

7.2.3 *Software* presente na plataforma TINI

Na figura 7.12 é apresentado um fluxograma do *software* implementado na TINI para este sistema. No início o programa da TINI fica bloqueado à espera da configuração do tempo de amostragem, esta configuração é feita pelo MATLAB via Ethernet. De seguida inicia-se o controlo. Em cada ciclo de controlo é medido o valor de resistência apresentado pelo sensor Watermark e é convertido para um valor de humidade. Com base neste valor é efectuada uma consulta à tabela difusa por forma a obter qual a duração do período durante o qual se deverá regar. Se a duração for nula não há actuação e regressa ao estado inicial até novo início de ciclo de controlo. Caso a duração do período de actuação não seja nula a bomba é activada até ao fim do mesmo, regressando ao estado inicial.

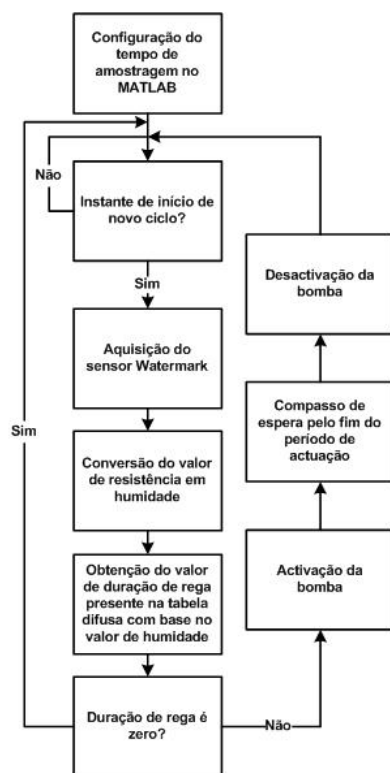


Figura 7.12: *Software* implementado na TINI

7.2.4 Resultados

Os resultados obtidos do controlo difuso podem ser observados no gráfico apresentado na figura 7.13.

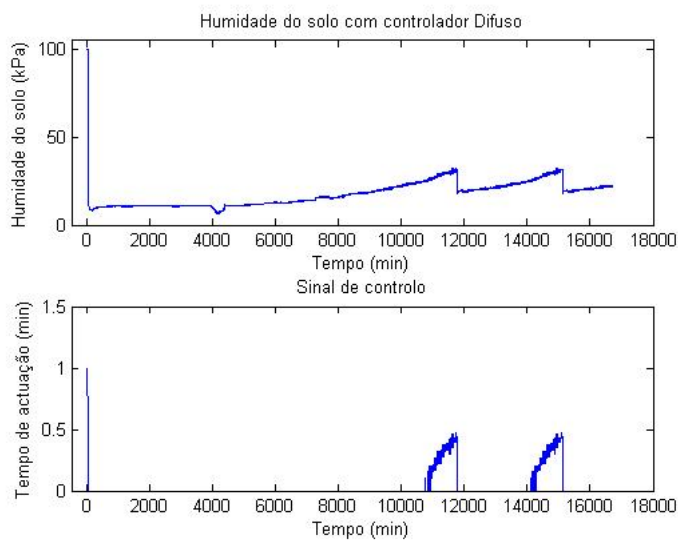


Figura 7.13: Humidade do solo - teste

Pela análise do gráfico conclui-se que o tempo de reacção ao início é bastante superior ao esperado. Apresenta um valor de 32 minutos desde o início do teste até a primeira transição do estado seco para o estado húmido, tal como pode ser observado em detalhe no *zoom* presente na figura 7.14, efectuado com base na figura 7.13. Isto é justificado pelo facto de o caudal em causa ser muito reduzido e de neste caso o sensor se encontrar inserido no solo.

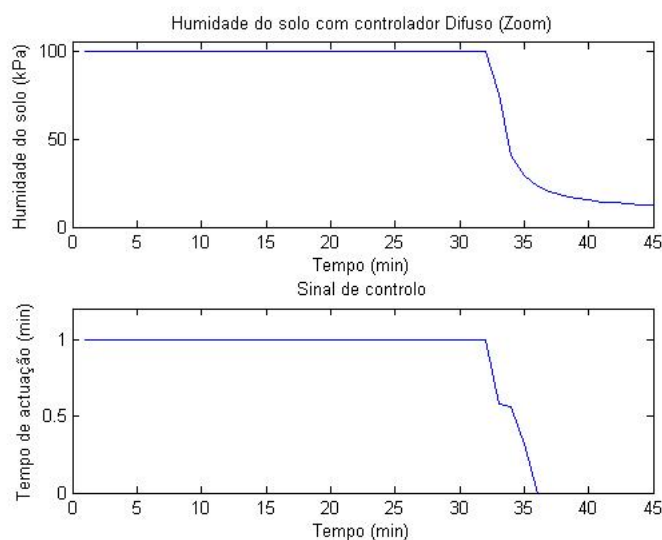


Figura 7.14: Humidade do solo - teste (*zoom*)

Conclui-se, adicionalmente, que a actuação efectuada ao início foi excessiva, uma vez que seria esperado que a humidade do solo ficasse próxima dos 26kPa (valor a partir do qual o controlador deixa de regar). Isto é justificado por um período de amostragem muito curto, levando a que o sistema actue muito mais rápido do que a a humidade varia. Outro motivo é o facto do sensor ter sido inserido no solo pouco tempo antes do início do teste, o que conduziu a uma percepção inicial errada da humidade do solo.

O decréscimo verificado no gráfico perto da amostra 4000 deve-se a um elevado aumento da temperatura, dado que durante esse período o vaso esteve directamente exposto ao sol. Mesmo nesta situação extrema a variação verificada não foi muito grande e se o sensor não se encontrasse num vaso, mas sim em terreno aberto a variação da humidade medida seria menor. Deste modo, tal como foi referido anteriormente este factor pode ser desprezado.

O sinal de controlo representa a fracção do período de amostragem durante o qual a rega deve ser efectuada. Observa-se que quando a humidade ultrapassa o limiar dos 26kPa, o sinal de controlo deixa de ser nulo. Verificou-se que mesmo após o início de actuação o valor medido do sensor indicava que o solo continuava no processo de

secagem, o que levou o controlador a aumentar o tempo de rega de acordo com a tabela 7.1. Quando a humidade desceu para um valor próximo de 20kPa, o controlador voltou a colocar o sinal de controlo a zero.

É possível verificar na figura 7.13 que ocorre um intervalo de tempo muito grande entre o início da actuação do sinal de controlo e uma variação na humidade em consequência desta. Isto deve-se ao facto do tempo que ocorre entre o início de rega e o humedecimento efectivo do sensor ser relativamente grande além disso o actuador não estava colocado perto do sensor, como na fase inicial, o que levou a que o sensor demorasse mais tempo a reagir. Verifica-se adicionalmente que quando há uma variação da humidade consequente da actuação esta é abrupta. Isto deve-se ao facto de o tempo de reacção na transição do estado seco para o estado húmido ser da ordem de grandeza do tempo de amostragem.

O teste teve um tempo de duração muito longo, uma vez que o solo não continha qualquer tipo de planta que absorvesse água, sendo o processo de evaporação muito lento.

A qualidade do controlo difuso está muito pendente da correcta parametrização dos conjuntos. Neste caso verifica-se uma oscilação do valor de humidade entre os valores de aproximadamente 20kPa e 30kPa. Este intervalo de variação poderia ser reduzido efectuando uma parametrização mais adequada. Adicionalmente poderia ser reajustado o tempo de amostragem para um controlo mais preciso bem como a posição relativa entre o actuador e o sensor.

Conclui-se, portanto, que é possível implementar um sistema de irrigação de baixo custo usando, no entanto, algoritmos de controlo bastante complexos, o que permite um controlo de qualidade com uma maior poupança de água.

Capítulo 8

Conclusões e Trabalho Futuro

Neste capítulo são tiradas algumas conclusões relativamente a este trabalho e são indicadas algumas linhas possíveis de trabalho futuro.

8.1 Conclusões

Ao longo deste trabalho foram apresentados vários tipos de irrigação, bem como os mais variados tipos de sensores e actuadores utilizados para o efeito. Foi realizada uma pequena descrição da linguagem Java e da sua utilização no âmbito de sistemas embutidos. Abordou-se a utilização de sensores de baixo custo do tipo *1-Wire*, sendo feita uma apresentação do seu modo de funcionamento. Adicionalmente, foi apresentada a plataforma TINI, elemento chave neste trabalho, bem como uma caracterização geral de controlo difuso.

Foi efectuada a implementação de variados tipos de circuitos de instrumentação visando o acondicionamento de sinal proveniente de sensores. Por fim, implementou-se com sucesso uma malha de controlo com base na utilização conjunta de linguagem Java, sensores *1-Wire*, controlo difuso e a plataforma TINI. Obtiveram-se alguns resultados e retiraram-se algumas conclusões.

Da realização deste trabalho conclui-se que podem ser implementados sistemas de irrigação autónomos de baixo custo mas com algoritmos de controlo complexos que permitem a poupança de água e, ao mesmo tempo, manter a qualidade do cultivo.

8.2 Trabalho futuro

Em perspectiva de trabalho futuro seria talvez importante a implementação de um acesso remoto via web sendo, para este efeito, necessária a criação de um servidor

na plataforma TINi. Criando um acesso deste tipo seria igualmente importante a implementação de um sistema de segurança, como por exemplo um sistema de *login* e *password* para que o sistema de irrigação não pudesse ser utilizado por terceiros.

Tendo em conta que se torna relevante a execução de testes para vários tipos de solos, plantas e sistemas de irrigação para optimização do algoritmo e como o processo a controlar é muito lento seria muito vantajosa a criação de um simulador, por forma a tornar esta tarefa mais simples e menos morosa.

Nesta linha de trabalho, pode-se ainda referir que seria útil a execução de testes em ambientes reais por forma a obter melhores caracterizações dos sistemas a controlar.

Bibliografia

- [1] *AICÁS & Eletrovalvulas* www.aicasa.cl.com.br Julho de 2009.
- [2] <http://java.sun.com/docs/books/tutorial/getStarted/intro/definition.html> Julho 2009.
- [3] Jantzen J. *Foundations of Fuzzy Control*. John Wiley & Sons, Ltd, 8 edition, 2007.
- [4] <http://jstik.systronix.com/compare.htm> Julho 2009.
- [5] Neto J. História e evolução da irrigação. Technical report, Informativo Verde, Edição 62, 2003.
- [6] Cudell G. *MANUAL DE INSTALAÇÃO DE REGA*. Gustavo Cudell, LDA., 2000.
- [7] Bernardo S., Soares A. A., and Mantovani E. C. *Manual de Irrigação*. Ed. Viçosa: Ed. UFV, 8 edition, 2006.
- [8] Segundo A. K. Desenvolvimento de um sistema de irrigação automático. Master's thesis, Ouro Preto, 2008.
- [9] Miranda J. H., Faria M. A., and Igarashi G. S. Irrigação localizada por mangueira plástica perfurada tripas.
- [10] Bernardo S., Mantovani E. C., and Palaretti L. F. *Irrigação princípios e métodos*. Ed. Viçosa: Ed. UFV, 2 edition, 2007.
- [11] PINTO P. R. *SISTEMA DE AUTOMAÇÃO DO IRRIGÂMETRO UTILIZANDO-SE INSTRUMENTOS DIGITAIS*. PhD thesis, VIÇOSA MINAS GERAIS BRASIL, 2008.
- [12] Wheather and soil moisture based landscape irrigation scheduling devices. Technical report, U.S. Department of the Interior Bureau of Reclamation Lower Colorado Region Southern California Area Office, 2007.

- [13] Fonseca I. M. Sensores, transdutores e detectores. Univ. São Judas (data desconhecida).
- [14] Werner H. Measuring soil moisture for irrigation water management. *SDSU College of Agriculture & Biological Sciences*, 2002.
- [15] Pinto C. A. R. and A.C. A. Borim. Medição de umidade no solo através de sensores capacitivos. *Revista de Ciências Exatas e Tecnologia*, 2006.
- [16] Valente A., Couto C., and Correia J. H. On-chip integrated silicon bulk-micromachined soil moisture sensor with electronics and bus interface. In *EFITA 2003*, 2003.
- [17] <http://www.ictinternational.com/images/10hs.jpg> Julho 2009.
- [18] Alam M. and Rogers D. H. Tensiometer used in scheduling irrigation. *Irrigation Management series*, 1997.
- [19] Smajstrla A. G. and Harrison D. S. Tensiometers for soil moisture measurement and irrigation scheduling. *University of Florida IFAS Extension CIR487*, 2002.
- [20] Zazueta F.S., Bucklin R., Jones P.H., and Smajstrla A.G. Basic concepts in environmental computer control of agricultural systems. *University of Florida IFAS Extension CIR1029*, 2008.
- [21] <http://www.ictinternational.com.au/images/2725.jpg> Julho 2009.
- [22] Tecnologias para gestão hídrica racional na olivicultura. *Irricampo*, (8), Janeiro 2006.
- [23] <http://www.mesasystemsco.com/images/products/Pico63.jpg> Julho 2009.
- [24] www.freepatentsonline.com Julho 2009.
- [25] www.madgetech.com Julho 2009.
- [26] *TECNOLOGIA ELECTROVÁLVULAS E VÁLVULAS Electroválvulas proporcionais ASCA/JOUVOMATIC* www.asca.com.br.
- [27] Stephen Glaudel. Ac vs dc determining the right power source for solenoid valves. *MRF Technical Spotlight*, 2006.
- [28] *Solenoids How to select your BLP Solenoid*, Dialight BLP Ltd.

- [29] Alecrim E. Máquina virtual java (java virtual machine). 2005.
- [30] Reily D. and Reilly M. *Java Network Programing And Distributed Computing*. A., Wesley, 2002.
- [31] Skehill O. Java in an embedded enviroment: Final year project report. Master's thesis, University of Limerick, 2004.
- [32] Jan Axelson. *Embedded ETHERNET AND INTERNET COMPLETE Designing and Programming Small Devices for Networking*. Lakeview Research LLC, 2003.
- [33] *DSTINIm400 Networked Microcontroller Evaluation Board Dallas Semiconductor* www.maxim-ic.com Julho 2009.
- [34] *DSTINIs400/DSTINIs-00x Sockets Evaluation Board Dallas Semiconductor* www.maxim-ic.com Julho 2009.
- [35] Edna Mie Kanazawa. *Estudo da aplicabilidade da plataforma Java em sistemas embarcados*. PhD thesis, Campo Montenegro São José dos Campos, 2005.
- [36] Neto A. Avaliação do sistema 1-wire para aquisição de dados de temperatura em instalações agrícolas. Master's thesis, Universidade Federal de Viçosa, 2003.
- [37] *Application Note 132 Quick Guide to 1-Wire net Using PCs and Microcontrollers Dallas Semiconductor*.
- [38] Neto A. J. S., Zolnier S., Lopes D. C., Martins J.H., and Filho A.J. Monitoramento remoto de temperatura utilizando o sistema 1-wiretm: Avaliação da transmissão de dados. *Exacta*, 3, 2005.
- [39] *MicroLAN In The Long Run Dallas Semiconductor* www.maxim-ic.com Julho 2009.
- [40] *Application note 148 Guidelines for Reliable Long Line 1-Wire® Networks* www.maxim-ic.com Julho 2009.
- [41] <http://para.maxim-ic.com> Julho 2009.
- [42] Mota A. M. and Fonseca J. A. Controlo fuzzy baseado em tabela. 2002.
- [43] Piedade M. B. Controlo difuso e a sua pertinência no caso linear: Algumas conclusões experimentais. Master's thesis, Universidade de Coimbra, 1998.

- [44] Roger Jang J.S. and Gulley N. *Fuzzy Logic TOOLBOX For Use with MATLAB*. Math Works Inc, 1995.
- [45] www.honeywell.com/sensing julho de 2009.
- [46] Berkeley St. Smx electrical interface for watermark or gypsum block sensor www.irrometer.com julho 2009. EME Systems.
- [47] www.mathworks.com/products/fuzzylogic julho de 2009.
- [48] Sensores da humidade dos solos irriometer sensores de humidade watermark. Nova Rocha.
- [49] Jantzen J. *Fuzzy Control*. Technical University of Denmark no 9109, 1991.

Apêndice A

Lógica Difusa

Os Conjuntos Difusos e a Lógica Difusa estão na base de técnicas para a solução de problemas, com uma grande aplicabilidade, especialmente, nas áreas de controlo e tomada de decisão. Destacam-se inúmeros exemplos de aplicações em diversas áreas como:

- No ramo automóvel, com as transmissões automáticas, ABS e o cruise control;
- No processamento de imagem, em câmaras, na realização de filmes e no reconhecimento de padrões;
- No desenvolvimento de inteligência artificial, em robótica e em vídeo jogos.

O ponto forte da Lógica Difusa deriva da sua capacidade de tirar conclusões e gerar respostas baseadas em informações qualitativas, vagas e ambíguas. Neste aspecto, os sistemas de base difuso são extremamente poderosos tendo a capacidade de raciocinar de forma semelhante à dos humanos. A teoria de conjuntos Difusos foi introduzida na década de 70 por L.A. Zadeh e A. Mamdani. Esta teoria pretende traduzir de forma matemática o conhecimento qualitativo adquirido pelos seres humanos com a sua experiência e torna-se óptima para sistemas de difícil modelação e não lineares. A Lógica Difusa é baseada na teoria dos conjuntos Difusos. Esta vem da teoria dos conjuntos clássicos para resolver os paradoxos gerados na classificação “verdadeiro ou falso” da Lógica Clássica. Normalmente, uma proposição lógica pode ser classificada de duas formas completamente opostas uma da outra: ou “completamente verdadeiro” ou “completamente falso”. Na Lógica Difusa isto não acontece, uma premissa varia em grau de verdade de 0 a 1, o que leva a existência de premissas parcialmente verdadeira ou parcialmente falsa. Com a teoria dos conjuntos Difusos passa a existir o conceito de “grau de verdade”. Os conjuntos são formados de forma qualitativa, ou seja, são

utilizados termos linguísticos qualitativos tais como: frio, grande, alto, perto, etc., os elementos destes conjuntos são caracterizados pelo grau de pertinência. Por exemplo, se a temperatura numa sala for de 13°C e a temperatura noutra sala for de 10°C, ambas podem ser consideradas membros do conjunto “frio”, no entanto, a sala com 10°C tem um grau de pertinência maior neste conjunto. Estes sistemas tornam a representação de comportamentos muito simples e natural, levando à construção de sistemas compreensíveis e de fácil manutenção. Informações deste capítulo baseadas em [3], [43], [49] e [44].

Teoria de Conjuntos

Como foi referido anteriormente, os conjuntos Difusos derivam da teoria dos conjuntos clássicos, tornando-se mesmo uma extensão a estes. Assim, para uma melhor compreensão de conjuntos difusos será primeiro exposto um pouco de teoria sobre conjuntos clássicos.

Conjuntos Clássicos

Um conjunto clássico, pode ser definido de várias formas. Por exemplo, se A for um conjunto, este pode ser definido como:

- $A = \{a_1, a_2, a_3\}$ onde a_1, a_2, a_3 são os elementos deste conjunto;
- $A = \{x \mid P(x)\}$ onde x tal que, x tem a propriedade de P , ou seja, todos os elementos de x para qual a proposição $P(x)$ é verdadeira;
- $\gamma_A(x) = \begin{cases} 1 & \text{para } \theta \in A \\ 0 & \text{para } \theta \notin A \end{cases}$ onde γ_A é a função característica do conjunto A , que define que elementos pertencem ou não ao conjunto.

Estes são portanto, conjuntos que se definem com um grau de veracidade extrema, ou seja, com proposições completamente verdadeiras ou completamente falsas. Alguns exemplos deste tipo de conjuntos podem ser:

- O conjunto de números não negativos e menores que 5 - $\{0, 1, 2, 3, 4\}$;
- O conjunto dos números naturais, $N - \{1, 2, 3, 4, \dots\}$;
- O conjunto dos números ímpares - $\{1, 3, 5, 7\}$;
- O conjunto de dinossauros vivos - ϕ (conjunto vazio).

Um conjunto pode fazer parte de outro, $A = \{1, 2\}$ pertence ao conjunto $B = \{\{1, 2\}, \{3, 4\}\}$, e um conjunto pode ser igual a outro desde que todos os seus elementos sejam os mesmos, por exemplo $A = \{1, 2, 3\}$ é igual ao conjunto $B = \{3, 2, 1\}$ uma vez que contêm os mesmos elementos, o mesmo acontece com os seguintes conjuntos $A = \{1, 2, 3\}$ e $B = \{3, 2, 1, 1\}$ que são iguais pela mesma razão que o exemplo anterior[3].

Conjuntos Difusos

Os conjuntos Difusos, podem ser definidos de várias formas[43]:

- $\gamma_A(x) = f(x)$, através da função características;
- $A = \sum_{i=1}^m \left(\frac{\gamma_A(x_i)}{x_i} \right) = \left(\frac{\gamma_A(x_1)}{x_1} \right) + \dots + \left(\frac{\gamma_A(x_m)}{x_m} \right)$, no domínio discreto, onde $\gamma_A \in [0, 1]$, representa o grau de pertença de x em A ;
- $A = \int_U \left(\frac{\gamma_A(x)}{x} \right)$, no domínio contínuo, onde $\gamma_A \in [0, 1]$, representa o grau de pertença de $x \in U$ em A ;
- $A = \{x, \mu_A(x) \mid x \in U\}$, onde o conjunto relaciona cada elemento x com um grau de pertinência $\mu(x)$, que pode tomar qualquer valor real no intervalo $[0, 1]$.

Assim os conjuntos Difusos distinguem-se dos clássicos uma vez que estes apenas podem tomar o valor de 0 ou 1, tal como é possível verificar na Figura A.1.

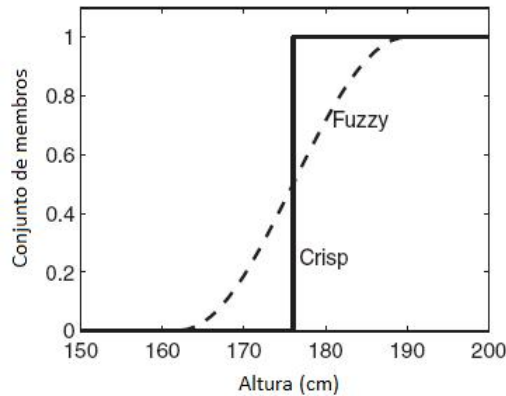


Figura A.1: Qualificação de “pessoas altas” segundo conjunto clássico (também conhecido como crisp) e conjunto difuso [3].

Definições básicas

Uma função de pertença contínua pode ser representada de duas formas: trapezoidal ou trapezoidal suave. Uma função pertença μ_A pode ser definida como a equação A.1.

Os parâmetros têm que ter a seguinte relação $a \leq b \leq c \leq d$, estes são os pontos do trapézio representados da esquerda para a direita [3].

$$\mu_A(x; a, b, c, d) = \begin{cases} 0 & , x \leq a \\ \frac{x-a}{b-a} & , a \leq x \leq b \\ 1 & , b \leq x \leq c, x \in \mathbb{R} \\ \frac{d-x}{d-c} & , c \leq x \leq d \\ 0 & , d \leq x \end{cases} \quad (\text{A.1})$$

No caso, do trapezoidal suave as funções lineares entre $a \leq x \leq b$ e $c \leq x \leq d$ são substituídas por equações não lineares por exemplo uma equação com cos como no exemplo A.2. A representação triangular é um caso particular da representação trapezoidal, que é o caso de $b = c$.

$$\mu_A(x; a, b, c, d) = \begin{cases} 0 & , x \leq a \\ \frac{1}{2} + \frac{1}{2} \cos\left(\frac{x-b}{b-a}\pi\right) & , a \leq x \leq b \\ 1 & , b \leq x \leq c, x \in \mathbb{R} \\ \frac{1}{2} + \frac{1}{2} \cos\left(\frac{x-c}{d-c}\pi\right) & , c \leq x \leq d \\ 0 & , d \leq x \end{cases} \quad (\text{A.2})$$

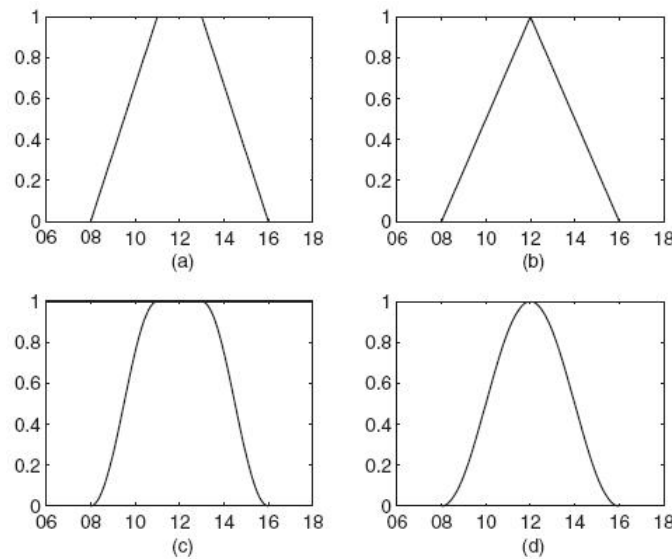


Figura A.2: Exemplo de funções de permanência μ_A (a)trapezoidal, (b)triangular, (c)trapezoidal suave, (d)triangular suave [3]

Operações de Conjuntos Difusos

Algumas das mais básicas operações de conjuntos são a igualdade e a inclusão. Considerando os conjuntos Difusos A e B pertencentes ao universo U , então estes serão iguais se as funções de pertinência forem iguais para todos os elementos A.3.

$$A = B \equiv \mu_A(x) = \mu_B(x) \quad (\text{A.3})$$

Considerando os mesmos conjuntos, A está incluído em B se a função de pertença de A for menor que a de B para todos os elementos A.4.

$$A \subseteq B \equiv \mu_A(x) \leq \mu_B(x) \quad (\text{A.4})$$

Existem outras três operações básicas de conjuntos: a união, a intersecção e o complemento. A união pode ser definida da seguinte forma

$$A \cup B \equiv \{x \mid x \in A \text{ ou } x \in B\}$$

$x \in A \cup B$ se existir algum elemento de x que pertença a A ou a B . A intersecção pode ser definido da seguinte forma,

$$A \cap B \equiv \{x \mid x \in A \text{ ou } x \in B\}$$

$x \in A \cap B$ se existir algum elemento de x que pertença a A e a B . O complemento pode ser definido da seguinte forma,

$$\overline{A} \equiv \{x \mid x \notin A\}$$

isto é, os elementos do universo que não pertencem ao conjunto A .

Estas mesmas operações em conjuntos Difusos podem ser definidas de uma forma um pouco diferente, a união Difusa A.5

$$A \cup B \equiv \{(x, \mu_{A \cup B}(x)) \mid x \in U \text{ e } \mu_{A \cup B}(x) = \max(\mu_A(x), \mu_B(x))\} \quad (\text{A.5})$$

a intersecção Difusa A.6

$$A \cap B \equiv \{(x, \mu_{A \cap B}(x)) \mid x \in U \text{ e } \mu_{A \cap B}(x) = \min(\mu_A(x), \mu_B(x))\} \quad (\text{A.6})$$

complemento Difuso A.7

$$\overline{A} \equiv \{(x, \mu_{\overline{A}}(x)) \mid x \in U \text{ e } \mu_{\overline{A}}(x) = 1 - \mu_A(x)\} \quad (\text{A.7})$$

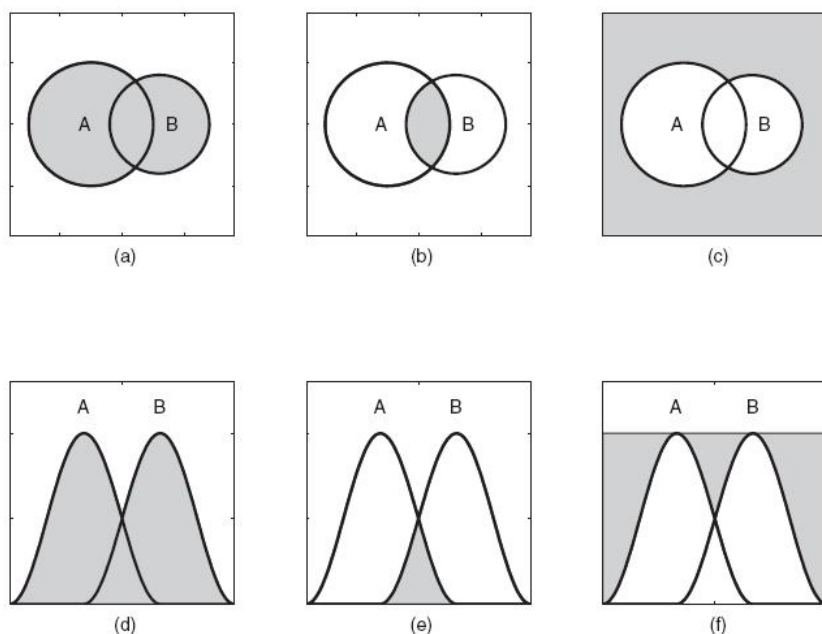


Figura A.3: União, intersecção e complemento. (a-c) conjuntos clássicos, (d-f) conjuntos difusos

Variáveis Linguísticas

Como referido anteriormente a lógica difusa utiliza variáveis linguísticas em vez de variáveis numéricas. Variáveis linguísticas são definidas com expressões e não com valores, por exemplo, “grande”, “pequeno”, “jovem”, etc. Estas variáveis podem alguns modificadores que alteram o significado destas e grau de pertença, por exemplo, “**muito** grande”, “**mais ou menos** jovem”.

Lógica Difusa

A lógica Difusa é baseada na teoria de conjuntos Difusos, tal como a lógica clássica é baseada nos conjuntos clássicos. Nesta secção serão abordados alguns conceitos básicos sobre a lógica Difusa.

Proposições Difusas e conectivas lógicas

A lógica Difusa, utiliza proposições Difusas, ligadas por conectivas lógicas para formar regras Difusas. As regras Difusas são a base do raciocínio Difuso.

Uma proposição Difusa pode ser por exemplo “ x é jovem”, onde jovem é uma variável linguística definida no universo de discurso da variável x , ou seja, as proposições Difusas estabelecem a ligação entre as variáveis Difusas pertencentes a um determinado universo e as variáveis linguísticas.

As conectividade lógicas mais utilizadas na lógica Difusa são a conjunção (*and*), a disjunção (*or*) e a negação (*not*).

Regras Difusas

As regras Difusas são do tipo If x Then y , no qual x e y são proposições Difusas. As regras Difusas podem ser formadas com proposições simples do tipo If x_1 é A_1 Then y é B , ou com proposições ligadas por conectivas lógicas do tipo If x_1 é A_1 and x_2 é A_2 Then y é B .

Propriedades das regras Difusas

Para um bom funcionamento as regras deve ter três propriedades: Consistência, Continuidade e Plenitude.

Consistência

As regras Difusas devem ser consistentes não devendo ter contradições entre elas.

Continuidade

As transições entre regras Difusas devem produzir respostas continuas de forma a não existirem saltos abruptos entre regras.

Plenitude

Cada entrada ou combinação de entradas deve ter sempre uma acção de saída.

Apêndice B

Instalação de *Software* TINI

Neste apêndice é efectuada uma descrição do processo de instalação do software necessário para a configuração de placas TINI via porta série usando o sistema operativo Windows. Neste será referida a instalação do *Java Development Kit* (JDK), *Java CommAPI*, *JavaKit* e como carregar o *firmware* da TINI.

Java Development Kit (JDK)

Este *software* é disponibilizado pela empresa Sun no seu site, podendo ser obtido em <http://java.sun.com/javase/downloads/index.jsp>. Neste projecto foi utilizada a versão Java SE Development Kit (JDK) Packet 6 Update 10. No entanto, o uso de outra versão, não deverá apresentar qualquer diferença ou dificuldade acrescida.

A instalação deste pacote é bastante simples bastando executar o ficheiro transferido e seguir os passos indicados, podendo eventualmente ser escolhido um directório destino diferente do pré-definido.

Após a instalação, é aconselhável adicionar às variáveis de sistema, o caminho para o directório onde está situado a compilador Java instalado, por forma a que este possa ser evocado sempre no directório actual, sem ter de referenciar a sua posição relativa no sistema de ficheiros. Para este efeito deverá seguir os seguintes passos:

Clicar com o botão direito do rato em *My Computer* e seleccione *Properties*. Na janela aberta, deverá seleccionar o menu *Advanced*, e carregar no botão *Environment Variables*. Selecciona a variável *path* e carregue em *edit*. No fim da linha, deverá adicionar a posição do directório (que contém o compilador) na hierarquia do sistema de ficheiros (neste exemplo em concreto C:\Program Files\Java\jdk1.6.0_10\bin\). Agora, o comando “javac” poderá ser evocado em qualquer directório.

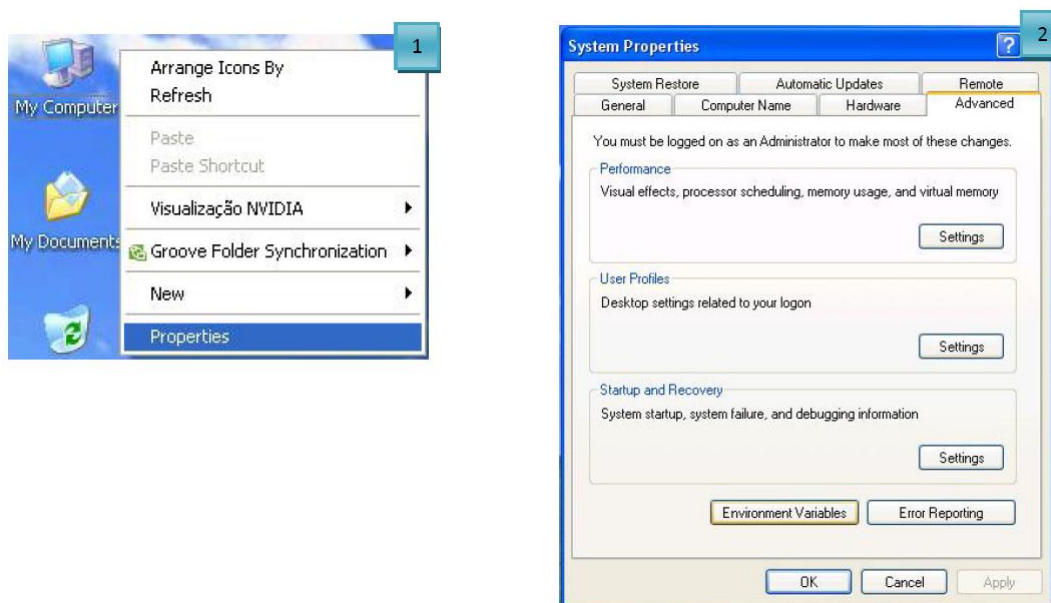


Figura B.1: Propriedades do sistema

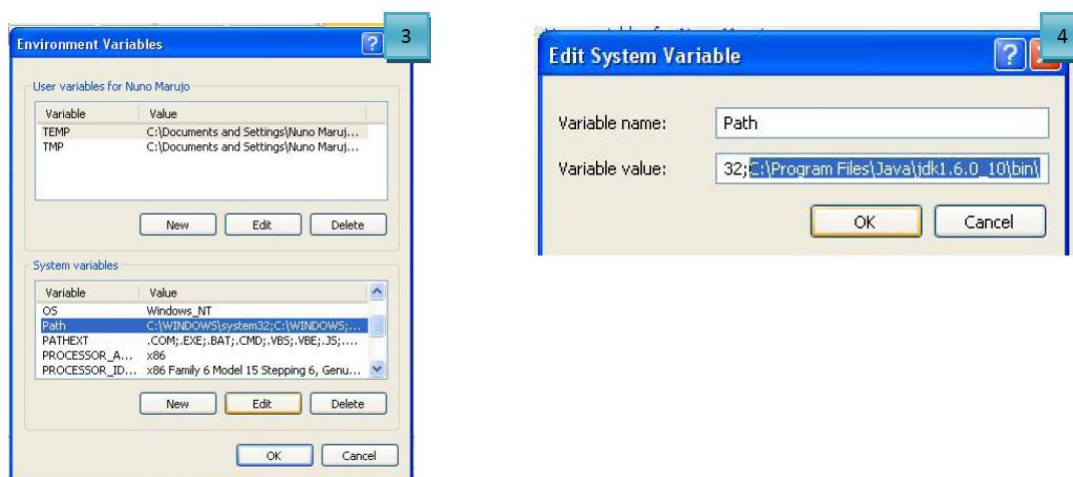


Figura B.2: Variáveis de ambiente

Java CommAPI e JavaKit

A última versão da *commAPI* lançada foi encontrada em:

<http://wind.lcs.mit.edu/download/java/javacomm20-win32.zip>.

O JavaKit pode ser encontrado em <ftp://ftp.dalsemi.com/pub/tini/> nos pacotes das duas versões de firmware disponíveis *tini1_02h.tgz* e *tini1_17.tgz*.

Para instalar o aplicativo de comunicação série *commAPI* e o loader *JavaKit* crie um directório por exemplo *C:\Javakit*. Descompacte o ficheiro *tini1_17.tgz*. Na

pasta `src`, encontrará o ficheiro `JavaKitSrc.jar`. Descompacte este ficheiro usando um aplicativo como, por exemplo, o Winrar e copie todo o seu conteúdo para a pasta criada anteriormente. De seguida, descompacte o ficheiro `javacomm20-win32.zip` e copie os ficheiros `comm.jar` e `javax.comm.properties`, incluídos neste, para a pasta criada anteriormente. Agora, basta abrir a linha de comandos, deslocar-se até à pasta que criou e correr os seguintes comandos (Nota: ter em conta que a sintaxe é *case sensitive*):

- `javac -cp "comm.jar;" *.java .` Este comando irá evocar o compilador, que fará a devida verificação de erros e linkagem dos ficheiros. (Nota: deverão aparecer dois warnings que não devem ser levados em conta);
- `java -cp "comm.jar;" JavaKit .` Este comando evocará o aplicativo `JavaKit`. Se até ao momento, seguiu todas as instruções, após a execução deste comando deverá visualizar a janela da figura B.3.

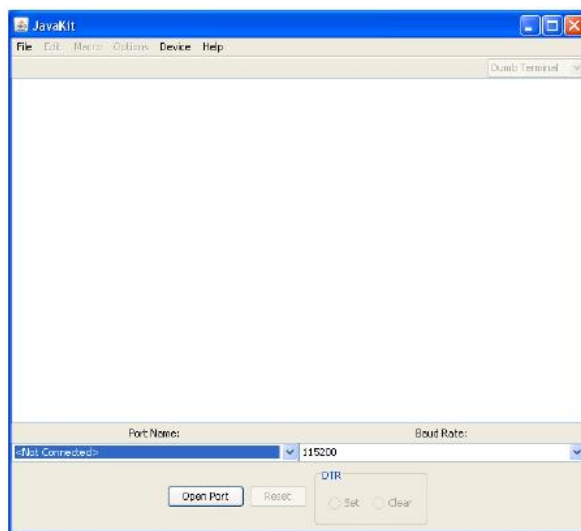


Figura B.3: JavaKit

De notar que o procedimento acima descrito, não é a única opção viável. É usual colocarem-se os ficheiros do aplicativo *commAPI* na pasta do *Java Runtime Environment* (JRE), no entanto, esta solução foi escolhida devido à sua portabilidade. Após o procedimento acima descrito, podemos facilmente transferir a pasta onde o aplicativo foi compilado para outro PC, e evocar o *JavaKit*, sem que haja qualquer quebra de *links*. Poderá, também, revelar-se uma ferramenta útil a criação de um ficheiro *batch* que evoque o segundo comando, criando assim, um executável que abrirá o *JavaKit* sem necessidade de o estar sempre a escrever.

Carregar *firmware* 1.1x no DS80C400 TINI

Para carregar o tini 1.1x Runtime Environment no DSTINIm400 ou em sistemas compatíveis devem ser carregados dois ficheiros `tini400.tbin` e `slush400.tbin` utilizando o JavaKit ou outro programa como o MTK. O ficheiro `tini400.tbin` carrega dados do endereço 400000h ao 470000h, e o ficheiro `slush400.tbin` carrega dados do endereço 470100h ao 480000h. Os 100h bytes entre o fim do `tini400.tbin` e o início do `slush400.tbin` são reservados para preservação das informações de configuração de rede (tais como o comando "ipconfig -C").

Assim para carregar os ficheiros basta ligar-se a TINI por porta com e ir ao menu do JavaKit, File\Load File e carregue os ficheiros `tini117-bank1to6.tbin` e `slush.tbin` (estes ficheiros encontram-se no directório "tini1_17\bin").

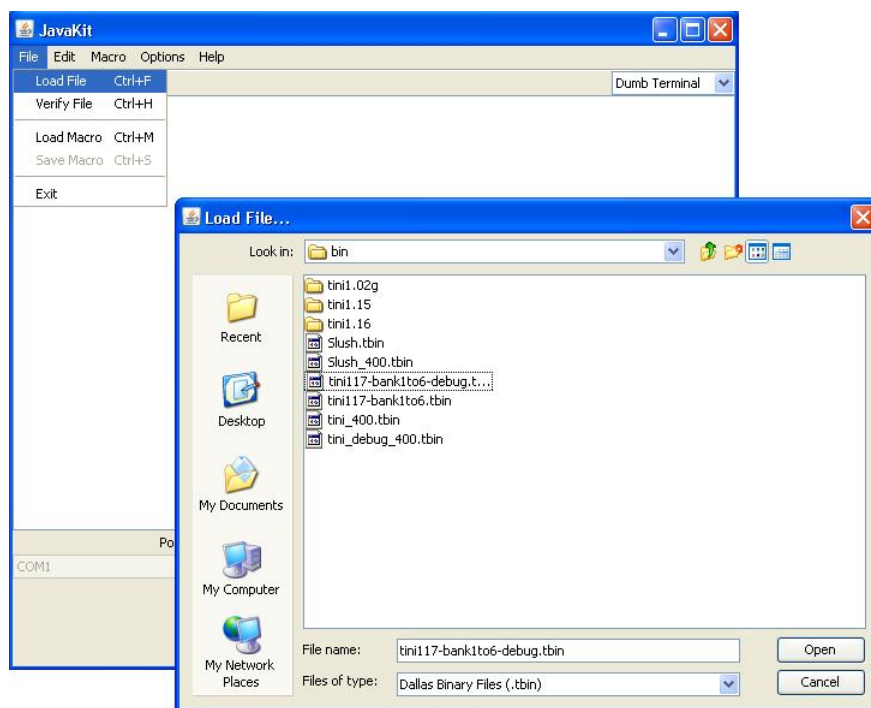


Figura B.4: Carregar firmware pelo JavaKit

De seguida é necessário escrever os seguintes comandos "B18", "F0" por forma a limpar a memória, e por último basta escrever o comando "E" para executar o novo *firmware*. Quando o *firmware* iniciar deverá aparecer na linha de comandos o seguinte (figuraB.5).

```

[==      slush Version 1.17      ==]
[      System coming up.      ]
[      Beginning initialization...  ]
[      Not generating log file.    ] [Info]
[      Initializing shell commands... ]
No response from TINI board!
[Done]

[      Checking system files...    ] [Done]

[ Initializing and parsing .startup... ]
[      Initializing network...    ]
[      Starting up Telnet server... ] [Done]
[      Starting up FTP server...   ] [Done]
[      Network configuration      ] [Done]
[      System init routines       ] [Done]

[      slush initialization complete. ]

Hit any key to login.

```

Figura B.5: Firmware 1.17

Apêndice C

Esquema Eléctrico do Interface Local

No âmbito desta dissertação foi desenvolvido um módulo de interface local. Na figura C.1 é apresentado o seu esquema eléctrico.

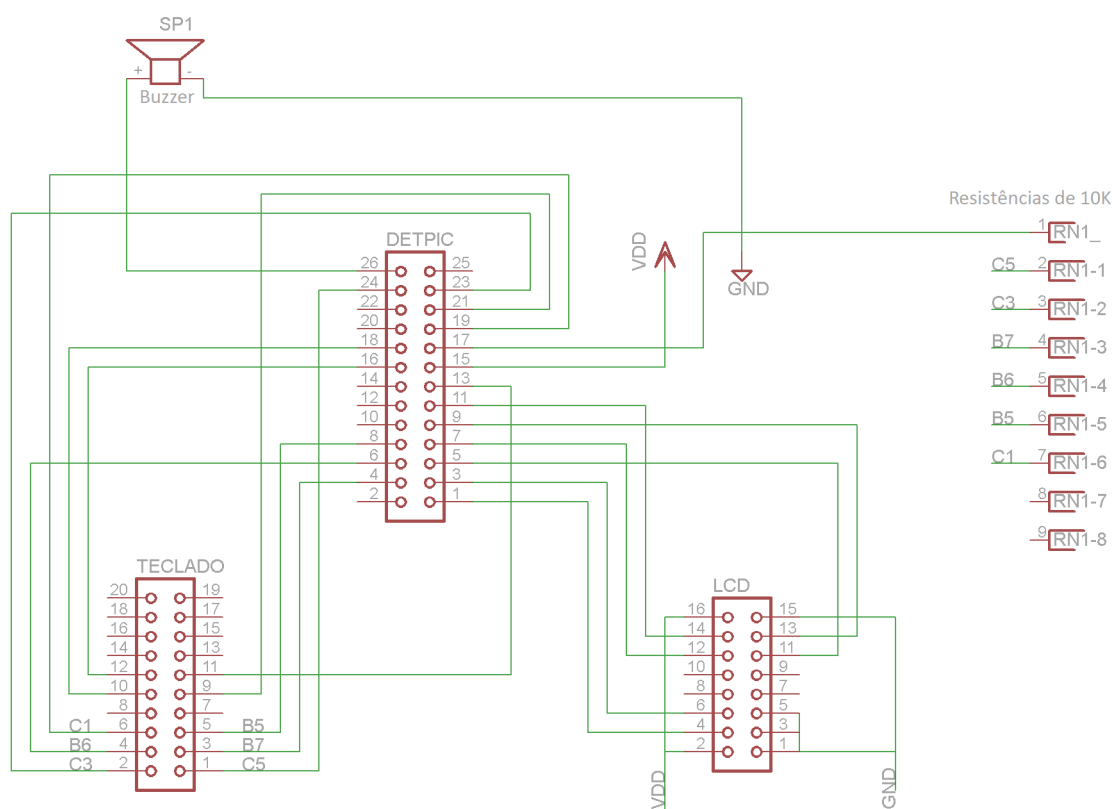


Figura C.1: Esquema eléctrico do interface local